

Tietokonekulttuurin erikoislehti



Pelit kulttuuri-
perintönä

Vim

Neuroverkko
uneksimaan

Kompuutterilla
kuumatkalle

Levytilaa
vanhaan
rautaan

- 
- 3 Pääkirjoitus**
- 4 Haastattelussa Mikko Tarmia**
Pelimusiikin ammattilainen kertoo.
- 8 MenuetOS**
Käyttöjärjestelmän saa nykyäänkin mahtumaan korpulle.
- 10 Tietokone typografian tuhoajana**
Tietokone ei taio taviksista graafisen alan ammattilaisia.
- 15 Kolumni: Mikko Heinonen**
Päivitimme laitteesi, se on nyt pilalla!
- 16 Otaniemen olohuone**
Koodia, kuvataidetta ja Skrolli-telkkaria.
- 20 Koneaivoa takomassa**
Kuinka neuroverkot toimivat, ja miten sellaisen voi rakentaa itse?
- 26 Vim**
Pieni ja elegantti tekstieditori.
- 32 Maasta Kuuhun – ja ehkä takaisin**
Ihminen koodasi itsensä kuuhun ja rupesi tekemään matkasta pelejä.
- 38 Lisää tallennustilaa Amigaan**
Modernit tallennusvälineet yhteistyöhön vanhan koneen kanssa.
- 46 Everdrive**
Pelimoduulit voi nykyään korvata Flash-muistilla.
- 49 Kolumni: Janne Sirén**
Nouseeko Nokia vielä Microsoftin tihutöiden jäljiltä?
- 50 Pelinkehityksen vallankumous**
Pelimoottoreiden vapautuminen on rikastuttanut indiepelien maailmaa.
- 55 Kirjoja**
Taidetta Kuusnepan ja Amigan kultakausilta Visual Compendium -kirjoissa.
- 58 Pelit kulttuuriperintönä**
Vaikka maailmanloppu iskisi, vanhat pelit pysyvät tallessa.
- 62 Ei näin!**
Super-Spectrumin nousu ja – arvatenkin – tuho.
- 64 Satunnainen samooja**
Nordicin turinat roolipelien paikallishistoriasta jatkuvat.
- 68 Yhden illan projekti**
Rakensimme joystickkejä diagnosoivan apuvälineen.
- 71 Mikrokivikausi**



Mikko Heinonen
asiantuntijatoimittaja

Skrolli

Tietokonekulttuurin erikoislehti

Yhteydenotot toimitus@skrolli.fi
Ircnet: #skrolli
skrolli.fi

Päätoimittaja Ville-Matias Heikkilä
Toimituspäälliköt Toni Kuokkanen & Ninna Koskenalho
Taiteellinen johtaja Risto Mäki-Petäys
Kuvatoimittaja Mitol Meerna
Mediamyynti Jari Jaanto
Talous Anssi Kolehmainen

Muu toimitus Mikko Heinonen, Jukka O. Kauppinen,
Ronja Koistinen, Teemu Likonen,
Annika Piironen, Manu Pärssinen,
Janne Sirén, Suvi Sivulainen

Tämän numeron avustajat David Carter, Ville-Veikko Heinonen,
Nordic the Incurable, Ville Jouppi,
Jukka K. Korpela, Elli Kurkikangas,
Antti Kurkinen, Essi Laurila,
Tapio Lehtimäki, Miikka Lehtonen,
Sakari Leppä, Teemu Purhonen,
Ville Ranki, Outi Sinisalo, Mikko Tarmia,
Mikko Torvinen, Mikko Tuomela,
Tomi Väisänen, Wallu

Julkaisija Skrolli ry

Painopaikka Tammerprint, Tampere,
ISSN 2323-8992 (painettu)
ISSN 2323-900X (verkkajulkaisu)

Skrollia kansalle!

Lehtemme voisi olla vielä enemmänkin.

Sinä voit auttaa siinä.

Nettihuumorista syntyneen lehtemme kolmas vuosikerta on kohta täynnä. Paperilehden tekeminen aikana, jolloin lehtitalojen yt-ruletti pyörii melkein joka viikko, on ollut todellista vastavirtaan uimista. Silti me olemme polskutelleet aika mukavasti.

Meillä on noin 2 000 tilaajan uskollinen yleisö, joka on sietänyt kiitettävästi muun muassa huonoa verkkokauppaamme ja laskutuksemme sekoiluja. Nyt meillä on lopulta tarjota nykyaikainen (ja useimmiten toimiva) verkkosivusto tilausjärjestelmineen. Tilauksen laskutuskin sujuu toivottavasti jälleen paremmin ensi vuonna.

Pari tuhatta tilaajaa on hyvä määrä, jolla lehteä voidaan julkaista, kun toimitustyö tehdään enimmäkseen harrastuspohjalta. Olemme kuitenkin varmoja, että sisällöstämme olisi iloa vielä useammallekin. Assembly-tapahtumassa tehty markkinointi paljasti jälleen, että moni innokaskaan tietotekniikan harrastaja ei ole vielä tietoinen Skrollista.

Jossain vaiheessa käy nimittäin väistämättä niin, että parhaankin lehden tekeminen alkaa maistua työltä. Suurempi tilaajamäärä mahdollistaisi sen, että voimme sitouttaa ihmisiä toimitustyöhön ja taata lehden jatkumisen myös sen jälkeen, kun alkuinnostus alkaa hiipua.

Toimitus on pohtinut asiaa palaverissaan jo pitempään, ja meillä on valmiina suunnitelmia, joiden avulla Skrollin tunnettuutta parannetaan. Tänä syksynä lehden mainoksia näkynee enemmän eri paikoissa, ja jatkamme myös irtonumeromyyntin kehittämistä. Onpa keskustelussa ollut mukana jopa kansainvälisen numeron tuottaminen. Linjastamme emme aio tinkiä tai ryhtyä julkaisemaan sähköverkkovertailuja – tämän tien ovat muut käyneet puolestamme.

Uskomme, että paras mainos Skrollille on lehti itse. Monet lukijoistamme ovat töissä it-alan yrityksissä ja tuntevat muutenkin alan harrastajia. Voitte edistää Skrollin asiaa kertomalla lehdestä tuttavillemme ja mainitsemalla, että vanhoja numeroita voi lukea verkosta ilmaiseksi. Jos haluatte näytelehtiä yrityksen kahvihuoneeseen, ottakaa yhteyttä toimitukseen. 📧

P.S. Haluamme jatkossa tarjota tilaajillemme enemmän etua. Siksi olemme muuttaneet kunkin numeron ilmaisen PDF-version julkaisuaikajakohtaa. PDF tulee jatkossa ilmaiseksi jakoon, kun seuraava numero lähtee painoon.



Etukannen kuva:
Tomi Väisänen



441 878
Painotuote



Haastattelussa Mikko Tarmia "Peli ilman musiikkia on kuin kossu ilman vissyä."

Hyvä pelimusiikki vahvistaa parhaillaan immersiota, peliin uppoutumista. Loistavaa pelimusiikkia ei välttämättä edes huomaa, jos se on toimivaa. Kansainvälisesti tunnettu mikkeliäinen muusikko Mikko Tarmia tietää kaiken tästä, sillä mies on säveltänyt esimerkiksi universaalisti rakastettuun Amnesia: The Dark Descent -peliin ääniraidan.

Teksti: Teemu Purhonen Kuvat: Teemu Purhonen, Mikko Tarmia

Mikko Tarmia kiinnostui musiikin tekemisestä vuonna 1990, jolloin hän teki tracker-ohjelmilla teknoa. Harrastus jatkui vuosien ajan. Vuosikymmenen loppupuoliskolla hän innostui erilaisten soittimien soittamisesta, ja sitä kautta musiikkityyli muuttui elektronisesta instrumenttimusiikkiin ja näiden välimaastoon. Videopelit ovat olleet aina osa hänen elämäänsä, ja pelimusiikin tekeminen tuntuikin luonnolliselta jatkumolta.

Aivan aluksi projekteja ei ollut juurikaan tarjolla, joten Tarmia ryhtyi harjoittelemaan orkesterimusiikin säveltämistä. Ensimmäinen projekti, sukellusvenepeli Deep Trouble Macille, osui kohdalle vuonna 2002. Sen jälkeen töitä onkin riittänyt niin pelien parissa kuin muutenkin, kuten Teemu Purhonen haastattelussa selvittää.

Kuinka peleihin säveltäminen eroaa muun musiikin työstämisestä?

Pelimusiikki toimii pitkälti samalla tavalla kuin elokuvamusiikki, eli sen funktio on tukea kuvaruudulla tapahtuvia tilanteita ja toimia osana tarinankerrontaa. Pelimusiikki eroaa kuitenkin elokuvamusiikista siinä mielessä, että se ei ole lineaarista vaan interaktiivista eli se on usein sidoksissa pelaajan toimintaan.

Interaktiiviselle musiikille on ominaista toisto, joten musiikin on kestävä toisteisuutta. Hyvin tehty musiikki antaa pelille tunnelman, tekee kokonaisuudesta saumattoman ja tällä tavoin syventää pelaajan immersiota eli uppoutuneisuutta pelin maailmaan.

Millainen prosessi pelimusiikin säveltäminen on kokonaisuutena?

Prosessi voi olla hyvinkin pitkä ja kestää pelin suunnitteluvaiheesta lähes julkaisuun asti – eli joskus useamman vuoden. Alkupäässä päätetään, minkä tyylistä musiikin tulee olla, ja valitaan sille alustavasti instrumentaatio. Pelin suunnittelija tai audiovasta-

va lähettää säveltäjälle listan tarvittavista musiikkiraidoista sekä kuvaukset siitä, mikä on minkäkin raidan funktio pelissä ja kuinka pitkiä niiden tulisi olla. Pelin sisältö saattaa muuttua vahvasti tuotannon aikana, joten musiikkiraidat tulee yleensä tehtyä aluksi hahmotelmina, jotka sitten tuotannon loppuvaiheessa viimeistellään valmiiksi.

Kuinka vapaat kädet säveltäjällä on peliprojektissa?

Riippuu aivan tilaajasta. Jos tilaajalla on erityinen käsitys siitä, miltä musiikin tulee kuulostaa, se rajoittaa tiettyä myös musiikintekijän vapauksia. Jotkut tarttuvat hyvinkin pieniin asioihin, mutta monesti myös säveltäjän vaistoon luotetaan. Pidän itse siitä, että joku vähän "piiskaa" säveltäjää, sillä se saa lopulta musiikin toimimaan paremmin pelissä. Eli vaikka musiikkiraita olisi muuten hyvä, voi säveltäjällä olla toisenlainen käsitys sen käyttötärpeestä kuin pelin suunnittelijalla.

Mikä pelimusiikissa on tärkeintä?

Tärkeintä on juuri se, että musiikki sulautuu peliin ja täydentää sitä. Joku peliarvostelija joskus sanoi, että hän ei edes huomannut pelissä olevaa musiikkia ja juuri se teki siitä erinomaista. Ajattelin, että tuohan on monessa määrin totta, mutta sen toimivuudessa tärkeää on harkinta musiikin käytön suhteen. Mitä, miten, milloin ja missä? Esimerkiksi tekemäni kauhupelimusiikki on luonteeltaan sellaista, että se toimii paljolti taka-alalla lähes piilossa mutta sille tulee aika ajoin aina tilaisuuksia nousta taustalta esiin.

Tämä mainittu ajattelutapa ei tietenkään päde kuin osaan peleistä. Jos otetaan vaikkapa Zelda-pelit, niin niissähän musiikki on lähes koko ajan sellaista, jota voi viheltää mukana. Ja se musiikki kestää myös toisteisuutta. Siinä on esimerkki erittäin hyvin tehdystä musiikista.

Pelimusiikissa parasta on, että se luo pelille tunnelman. Vaikeinta on ehkä sellaisen yhtenäisyyden rakentaminen, joka kulkee ja kehittyy pelin alusta loppuun.

Kuinka yhteistyösi Frictional Gamesin (Penumbra, Amnesia: The Dark Descent, SOMA) kanssa sai alkunsa?

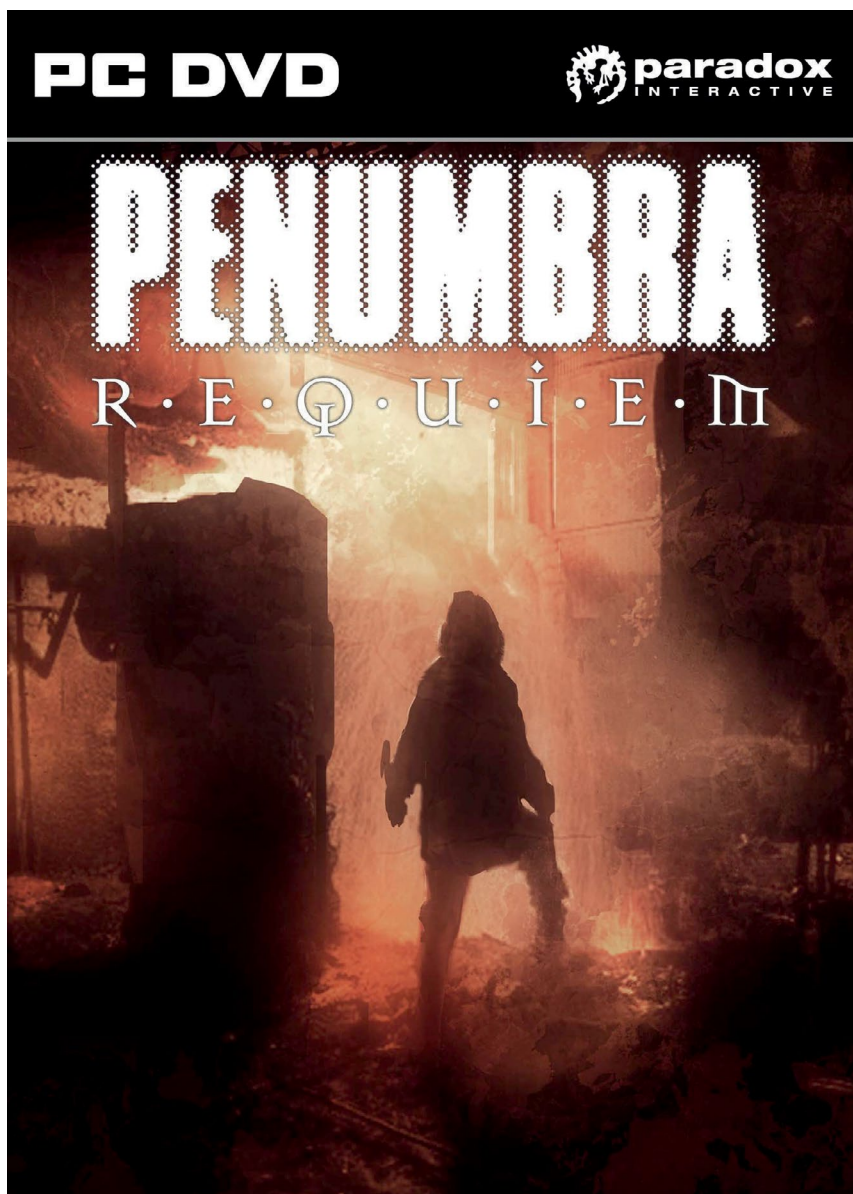
Olin samoissa peliprojekteissa ruotsalaisen äänisuunnittelijan Jens Nilssonin kanssa. Hän oli jo aikaisemmin pyytännyt minua tekemään musiikkia tekniseen demoon, jota he olivat kaverineen työstämässä. Demo oli Penumbra, joka sai netissä niin hyvän vastaanoton, että siitä päätettiin tehdä kaupallinen peli. Frictional Games -yhtiö syntyi ja vaikeuksien kautta nousi yllättäen indiepeliyhtiöiden kärkisarjaan. Olen tehnyt heidän kanssaan töitä lähes alusta asti, vuodesta 2006.

Millaisella laitteistolla tuotat musiikkia?

Tärkein työkaluni musiikin tuottamisessa on Logic Studio -ohjelmisto, jota olen käyttänyt jo pitkälle toistakymmentä vuotta. Siinä ohessa on käytössä suuri määrä erilaisia sämpelikirjastoja sekä muita lisäosia. Kaikkea ei kuitenkaan tehdä tietokoneella, vaan studiosta löytyy myös valikoima erilaisia syntikoita ja akustisia instrumentteja. Lisäksi löytyy myös perinteistä studiorautaa eli kompressoreita



Säveltäminen on säveltämistä, teki sitä sitten koneella tai akustisella kitaralla. Tarmia hallitsee-kin sekä konemusiikin että aidot instrumentit.



Pienestä teknologiademosta kasvoi kulttipeli, joka on ruokkinut kauhunnäikäisten pelokertoimia jo lähes vuosikymmenen ajan.

ja efektejä. Syntikoita löytyy muun muassa pieni modulaarinen setti, Nord Rack 2, Korg MS-20, DSI Mopho sekä hieman erikoisempi Akai EWI, joka on elektroninen puhallininstrumentti analogimoduuliin kytkettynä.

Kenen pelisäveltäjän musiikkia arvostat eniten?

Pitkän linjan säveltäjistä Koji Kondo (Zeldat, Mariot) olisi varmaankin se ykkönen, jos joku pitäisi valita. Yksittäisistä uudemmissa tuotoksista on painunut mieleen Austin Wintoryn tekemä ääniraita Journey-peliin. Se on myös ensimmäinen Grammy-ehdokkuuden napannut pelisoundtrack.

Suhtaudutaanko pelisäveltäjiin eri tavalla kuin säveltäjiin yleensä?

Jossain vaiheessa pelimusiikkia ei arvostettu samaan tapaan kuin esimerkiksi elokuvamusiikkia, mutta tilanne on vuosien varrella muuttunut. Pelibisnes on pyyhältänyt jo elokuvabisneksen ohitse, ja ihmiset käyttävät yhä enemmän aikaa pelaamiseen. Tunnetut elokuväsäveltäjät tekevät välillä musiikkia peleihin ja pelimusiikot puolestaan elokuvaan. Profiloitumisesta on tullut häilyvämpää. Vaikka minutkin profiloidaan pelisäveltäjäksi, näen itseni enemmän muusikkona, joka tekee musiikkia erilaisiin käyttötarkoituksiin. Pelit ovat yksi ilmaisun alusta, joka saavuttaa myös ihmiset hyvin.

Kuinka paljon itse pelaat?

Aina silloin tällöin, kun ehtii. Suosikkipelejäni ovat Zeldat, Extreme sports -tyyppiset pelit (Tony Hawkit yms.), RTS-pelit (Starcraft on yksi parhaista) sekä vanhat arcade-pelit.

Pystytkö pitämään pelistä, vaikka siinä sattuisi olemaan mielestäsi huonot musiikit?

Kyllä, jos peli on itsessään hyvä. Musiikit saa yleensä pois päältä. :-)

Millä tapaa on ollut erilaista säveltää musiikkia uusimpiin peliprojekteihisi, SOMAan ja Overgrowthiin?

Täysin erilaista. SOMAssa instrumentaatio on lähes kauttaaltaan syntetisistä, joten suuri osa ajasta menee instrumenttien luomiseen. Musiikin tekeminen SOMAan on enemmänkin äänimaisemien maalailua kuin varsi-



Ruotsalaisen Frictional Gamesin kehittämä, syyskuussa julkaistava SOMA on täyttä avaruus-kauhua.

naista säveltämistyötä. Overgrowthin instrumentaatio nojaa puolestaan perinteisiin instrumentteihin (lähinnä jouset, puhaltimet ja perkussiot), joten voin keskittyä puhtaasti säveltämiseen.

Millainen kokemus oli voittaa vuoden 2011 Independent Game Festivalissa Excellence in Audio -palkinto ääniraidasta peliin Amnesia: The Dark Descent?

Olihan se hienoa, että saimme työtämme sillä tavoin amatillista tunnustusta. En seurannut palkintogaalaa livenä, mutta sieltä sitten pamahti heti perään muilta sähköpostia, että palkinto tuli. En koe kuitenkaan, että palkinnon saaminen olisi ollut mitenkään merkittävää uusien projektien saamisen kannalta.

Oletko ollut ammatillisissa tekemisissä kotimaisten pelitalojen kanssa?

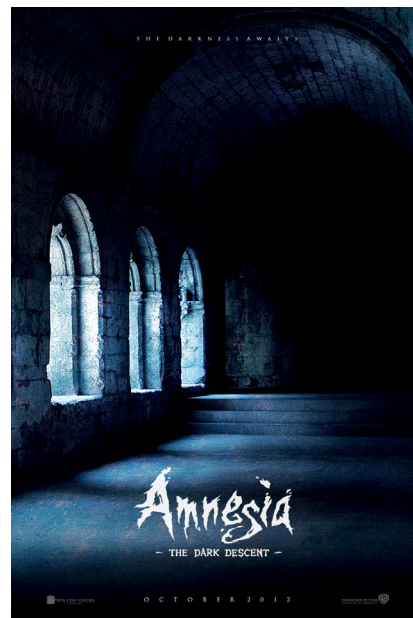
Hyvin vähän. Olen tehnyt ainoastaan yhdelle suomalaiselle pelitalolle pienemmän projektin. En tietenkään välttele kotimaisia pelitaloja, mutta Suomessa kysyntää työlleni ei ole vain ollut. Täällä on kuitenkin paljon osavia pelisäveltäjiä, joten tekijöitä riittää.

Onko pelisäveltäjän ura häirinyt muun musiikin tekemistä tai julkaisemista?

Onhan se. Aikaa ns. oman musiikin tekemiselle ei juurikaan ole ollut.

Kertoisitko levy-yhtiöstäsi The Sound of Fictionista?

The Sound of Fiction ei oikeastaan ole varsinainen levy-yhtiö, vaan

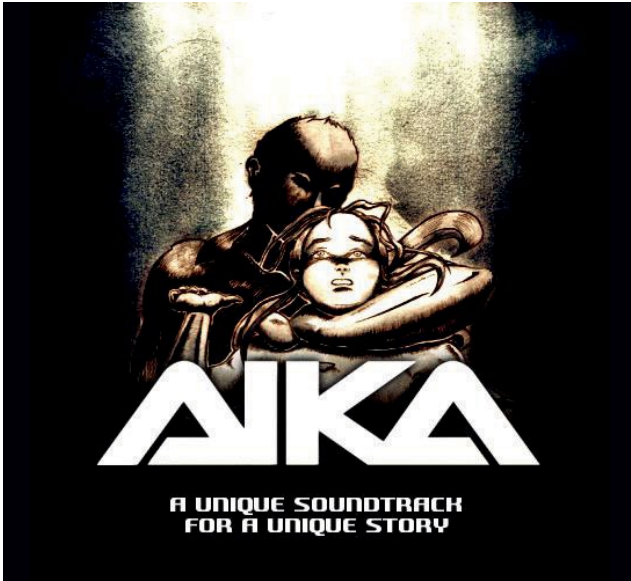


Kauhuseikkailu Amnesia on näyttänyt kiitosta, ei vähiten musiikkiansa ansiosta.

se on enemmänkin kanava omien tuotantojen julkaisemiseen, lähinnä pelisoundtrackien. Ensimmäinen julkaisu AIKA oli puolestaan peleistä irrallinen projekti, jossa teimme muutama säveltäjän voimin musiikkia, joka kuvaili levyä varten kirjoitetun novellin tapahtumia. Mukana projektissa oli myös kirjoittaja ja kuvittaja.

Tunkevatko popmusiikin trendit myös pelimusiikkiin? Onko tällä hetkellä nähtävissä jonkinlaisia trendejä pelimusiikissa, kuten esimerkiksi retrosaundeja, dubstepiä tai muuta?

Kyllä ne kantautuvat pelimusiikkiinkin. Varsinkin indiemarkkinoille on pullahtanut paljon retrohenkisiä pelejä, joissa vastaavasti ammennetaan



AIKA oli irtiotto peleistä.



Säveltäjän suurin unelma? Oma nimi pelin tai edes soundtrackin kannessa.

saman aikakauden saundimaailmaa. Pop-musiikissahan 80-luvun vaikutteita on ollut esillä jo pidemmän aikaa.

Onko olemassa peliä tai pelisarjaa, johon säveltäminen olisi urasi hienoin saavutus, oikea Graalin malja?

Ei oikeastaan. Olen erittäin otettu jo näistä mahdollisuuksista, joita minulle on tarjoutunut, ja uskon, että tulen tekemään jatkossakin mielenkiintoisia projekteja. En lähde aivan kaikenlaisiin projekteihin mukaan vaan arvioin sitä, onko pelillä mitään innovaatioarvoa pelialalle ja itselleni muusikkona.

Mikä on omista töistäsi se kaikkein tärkein? Miksi?

Kyllä se on Amnesia, ihan vain sen saavuttaman suosionkin vuoksi. Pelin soundtrackia jälkepäin analysoituaani tuntuu, että sain tehtyä musiikista jokseenkin eheän kokonaisuuden, joka istuu pelin visuaaliseen tyyliin hyvin. Musiikin luomisvaiheessa olin

skeptisempi onnistumisen suhteen, mutta tällainen epäily tuntuisi kuuluvan sävellysprosessiini jokaisen pelin kohdalla.

Tunnetko paljon muita pelisäveltäjiä? Onko esimerkiksi netissä pelimusiikkisäveltäjien yhteisöä?

Täytyy sanoa, että en tunne kovinkaan montaa pelisäveltäjää. On olemassa ainakin sellainen isompi järjestäytynyt yhteisö kuin G.A.N.G. (Game Audio Network Guild), mutta en ole kokenut tarvetta liittyä mihinkään sellaiseen.

Millaisia vinkkejä antaisit pelimusiikkialalle tähtäävälle?

Jotta olisi valmiuksia ottaa vastaan hyvin erityyppisiä projekteja, kannattaa harjoitella musiikin tuotantoa laaja-alaisesti aina orkesterimusiikista elektroniseen musiikkiin ja jopa pop- ja jazz-musiikkiin, sillä koskaan ei tiedä, mitä tilaaja haluaa. Projektien saaminen on paljolti suhdetoimintaa, joten olisi hyvä päästä tutustumaan alalla työskenteleviin ihmisiin.

Projektien saaminen vaatii myös kärsivällisyyttä. Netistä löytyy useita pelikehittäjien keskustelupalstoja, joissa on välillä tarjolla töitä sä-

veltäjillekin. Ottajia on kuitenkin paljon, joten hyvän portfolion luominen auttaa työn saamisessa. Ja sitä mukaan kun tehdyt projektit lisääntyvät CV:ssä, sitä helpompaa on saada töitä. Alku on se hankalin vaihe.

Missä näet itsesi säveltäjänä viiden vuoden päästä?

Olen haaveillut saavani enemmän jalkaa oven väliin elokuvamusiikin puolelle, joten toivottavasti sillä saralla tapahtuu edes jotakin viidessä vuodessa. Toki tulen jatkamaan pelienkin parissa, mutta pieni vaihtelu ei olisi pahasta. 🐾

Mikko Tarmia (s. 1978)
Asuu Mikkelissä
Koulutus: Tampereen ammattikorkeakoulu
<http://www.mikkotarmia.com>

Videopelisävellykset

- Deep Trouble (2002)
- Epsilon Tahari: Reign of the Machines (2003)
- Rally Shift (2004)
- Deep Trouble 2 (2005)
- Penumbra: Overture (2007)
- Penumbra: Black Plague (2008)
- Penumbra: Requiem (2008)
- Amnesia: The Dark Descent (2010)
- Overgrowth
- SOMA

Palkinnot: Excellence in Audio -palkinto Independent Games Festival -gaalassa vuonna 2011.



Pitkään kehitetty indieseikkailu Overgrowth on sananmukaisesti eläimellinen peli.



MenuetOS – kurkistus minimalistiseen käyttöjärjestelmään.

Viisitoista vuotta harrastuksena kehitetty käyttöjärjestelmä MenuetOS saavutti keväällä versioon 1.0. Katsastimme järjestelmän ja haastattelimme projektin perustajaa Ville Turjanmaata.

Teksti: Ronja Koistinen Kuvat: Risto Mäki-Petäys, Ronja Koistinen

Vuodesta 2000 lähtien Ville Turjanmaa on luotsannut puhtaalta pöydältä aloitettua MenuetOS-käyttöjärjestelmäprojektia. Järjestelmä ei pohjautu esimerkiksi Unix- tai Posix-malleihin, vaan se on suunniteltu alusta asti itse. Se on myös kirjoitettu kokonaan x86-assemblerilla. MenuetOS saavutti tänä keväänä version 1.0, joten on korkea aika hieman tutustua siihen. Nappasimme version 1.02.20 ja panimme sen Virtualboxiin pyörimään.

Kysyimme kuitenkin aluksi Ville Turjanmaalta, mikä on saanut hänet perustamaan MenuetOS-projektin ja millaiseen tarpeeseen järjestelmä vastaa. Hän vetoaa muutta mutkitta konekieliohjelmoinnin ylivertaisuuteen:

– Konekieli on objektiivisesti tarkasteltuna muita ohjelmointikieliä tehokkaampaa ja nopeampaa. Ja osa ohjelmointikielistä on jo niin kaukana prosessorista, että ne muistuttavat ratkaisuiltaan enemmän humanistisia tieteitä.

32-bittinen MenuetOS on julkaistu GPL-lisenssillä ja on siten vapaata ohjelmistoa mutta myöhemmin uudelleen kirjoitettu 64-bittinen versio ei ole vapaasti saatavilla, mutta se on saatavilla ilmaiseksi, ja yksityinen, epäkaupallinen käyttö on sallittua.

MenuetOS on reaaliaikainen, useaa suoritinta tukeva, graafinen työpöytä-käyttöjärjestelmä. Se on hyvin nopea ja pieni: koko hoito mahtuu yhdelle 1,44 megatavun levykkeelle. Laitteistotuki on vielä suppea, mutta hyvällä alulla. Menuet tukee joitakin verkkokortteja, äänikortteja, tulostimia, webkamerointa, digi-tv-kortteja ynnä muuta. Kaikki grafiikkalaskenta tehdään keskussuorittimella, joten vaatimuksia monimutkaisille näytönohjainajureille ei ole.

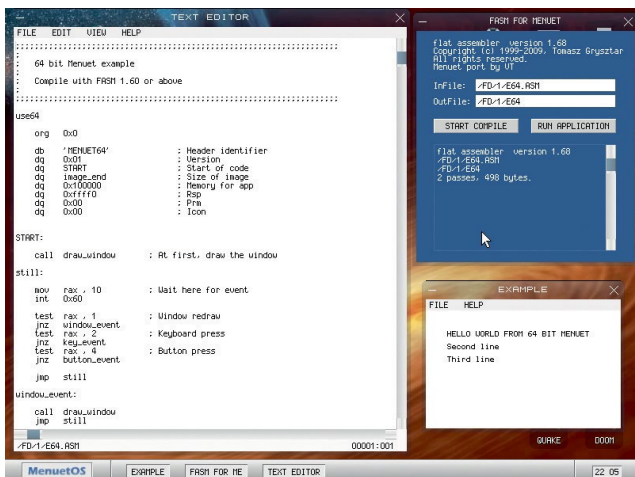
Kaikkea tätä Turjanmaa ei suinkaan ole yksin puurtanut kokoon.

– Menuetin kehittämiseen on osallistunut muun muassa Mike Hibbett (TCP/IP), Madis Kalme (GUI-funk-

tiota) ja Akos Mogyorosi (Fouriermuunnos ja audiodekoodereita). Lisäksi useita muita ihmisiä Euroopasta ja muualta, Turjanmaa valottaa kehitystiimin koostumusta.

Turjanmaa muistuttaa myös näytönohjainkorttien suljettujen rajapintojen aiheuttamista yhteensopivuusongelmista monissa järjestelmissä:

– Menuetissa on perusversiot useista käyttöjärjestelmästä odotetuista ominaisuuksista: esimerkiksi pre-emptive moniajo, moniprosessorituki ja graafinen käyttöliittymä läpinäkyvyyssefektillä. Ja koska konekieli mahdollistaa nopeamman koodin suorittamisen, niin graafisen käyttöliittymän laskutoimitukset tehdään 64-bittisessä x86-prosessorissa, mikä parantaa yhteensopivuutta. Useilla muilla käyttöliittymillä on ollut hankaluuksia nimenomaan grafiikkakorttien kanssa. Eikä tämä ilmeisesti lähiaikoina tule muuttumaan, sillä valmistajat pitävät tarkemmat API:t omana tietonaan.



MenuetOS:n mukana tulee FASM-konekielikääntäjä ja muutama esimerkkiohjelma lähdekoodeineen.



Työpöytä on perinteisen selkeä mutta menevän värinen.

Paljon säätämistä, niukasti ominaisuuksia

MenuetOS:ää kokeiltiin Virtualbox-virtuaalikoneella, jonka käyttöön on selkeät ohjeet käyttöjärjestelmän nettisivuilla. Ohjeissa esimerkiksi neuvotaan, millaisen verkkokortin emulointi kannattaa valita ja kuinka verkkoyhteys virtuaalikoneesta käsin pystytetään.

Käyttömukavuudessa Menuet on vielä lapsenkengissä. Nykypäivänä tuntuu vaivalloiselta pystyttää verkkoyhteys manuaalisesti ja määrittää järjestelmän ominaisuuksia naputtelemalla heksadesimaaleja valtavaan asetustiedostoon. Esimerkiksi tuki hiiren rullalle on oletuksena pois päältä, ja jos sitä kokee tarvitsevansa, täytyy käydä asetustiedostossa vaihtamassa yksi 0x00 0x01:ksi. Samaten asetustiedostossa varataan järjestelmälle oletuksena niin vähän keskusmuistia, että web-selain kaatuu ilmoitukseen liian pienestä muistimäärästä.

Ehkäpä asetusten määrittelyyn on kuitenkin myöhemmin tulossa automaatiota. On toki ymmärrettävää, että vielä ei ole moiselle riittänyt resursseja.

Linux-käyttäjälle Menuetissa tuntuu hassulta, että se on kokonaan graafinen. Testissämme virtuaalikone käynnistyy suoraan graafiselle työpöydälle, eikä mistään löydy minkäänlaista komentoriviä. Jopa konekielikääntäjä FASM on yksinkertainen ikkunoitu ohjelma, jossa tekstikenttiin kirjoitetaan luettavan lähdekoodin ja ulos tulevan binääriin polut.

Hyvä harjoitusprojekti voisikin olla koodata Menuetiin terminaaliemulaattori, Unix-tyyppinen komentotulkki ja työkaluohjelmat.

Minimalismissaan ja yksinkertaisuudessaan Menuet on kaikesta huo-

limatta valtavan viehättävä. Jopa projektin dokumentaatio ja nettisivut ovat pienet ja karut, mutta lyhyen hämmästelyn jälkeen ne osoittautuvat täysin kattaviksi. Esimerkiksi Menuetin kaikkien järjestelmäkutsujen dokumentaatio on aivan tavallinen 70 kilotavun tekstitiedosto, jossa yksinkertaisesti luetellaan yksi kerrallaan, miten kukin kutsu tehdään ja mitä se palauttaa. Herkullisen suoraviivaista.

Järjestelmäkutsuluettelon lisäksi projektin sivustolta löytyy toki myös linkkejä harrastajien foorumeille, pieniä esimerkkikoodin pätkiä ja selontekoa siitä, millä laitteilla järjestelmä on todettu toimivaksi. Siis esimerkiksi tieto toimivista verkko- ja äänikortteista löytyy sivustolta helposti.

Kuka tätä käyttäisi?

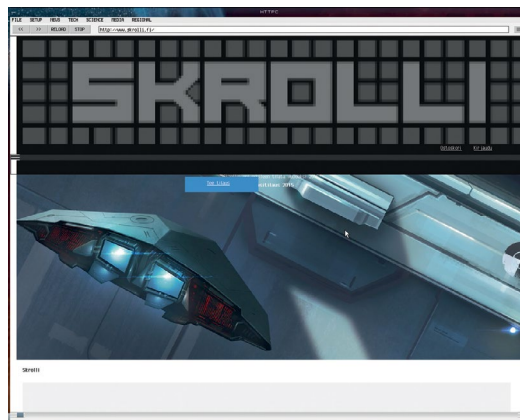
Järjestelmässä on helppolukuinen dokumentaatio ohjelmointiin nimenomaan assemblerilla, sillä Menuet on itsekin kirjoitettu sillä kokonaan. Samaan tapaan kuin unixeissa tulee mukana C-kääntäjä ja hyvät manuaalisivut C-ohjelmointiin, Menuetissa on tilalla valmiiksi asennettu FASM-konekielikääntäjä ja ohjeet konekieliohjelmointiin.

Menuet soveltuu siis erinomaisesti harjoittelualustaksi ohjelmointia jo jonkin verran taitavalle nikkarille – erityisesti sellaiselle, jonka sielua kalvaa nykyajan ohjelmistojen tuhlaileva laskentatehon käyttö. Kaveriksi voi ottaa Skrollin numeron 2015.2, jossa julkaisimme pienen johdatusjutun siihen, mistä assembler-kielissä on kyse.

Myös Turjanmaa itse pitää konekielipohjaisuutta sekä ohjelmointirajapinnan nopeaa omaksuttavuutta Menuetin tärkeänä puolena:

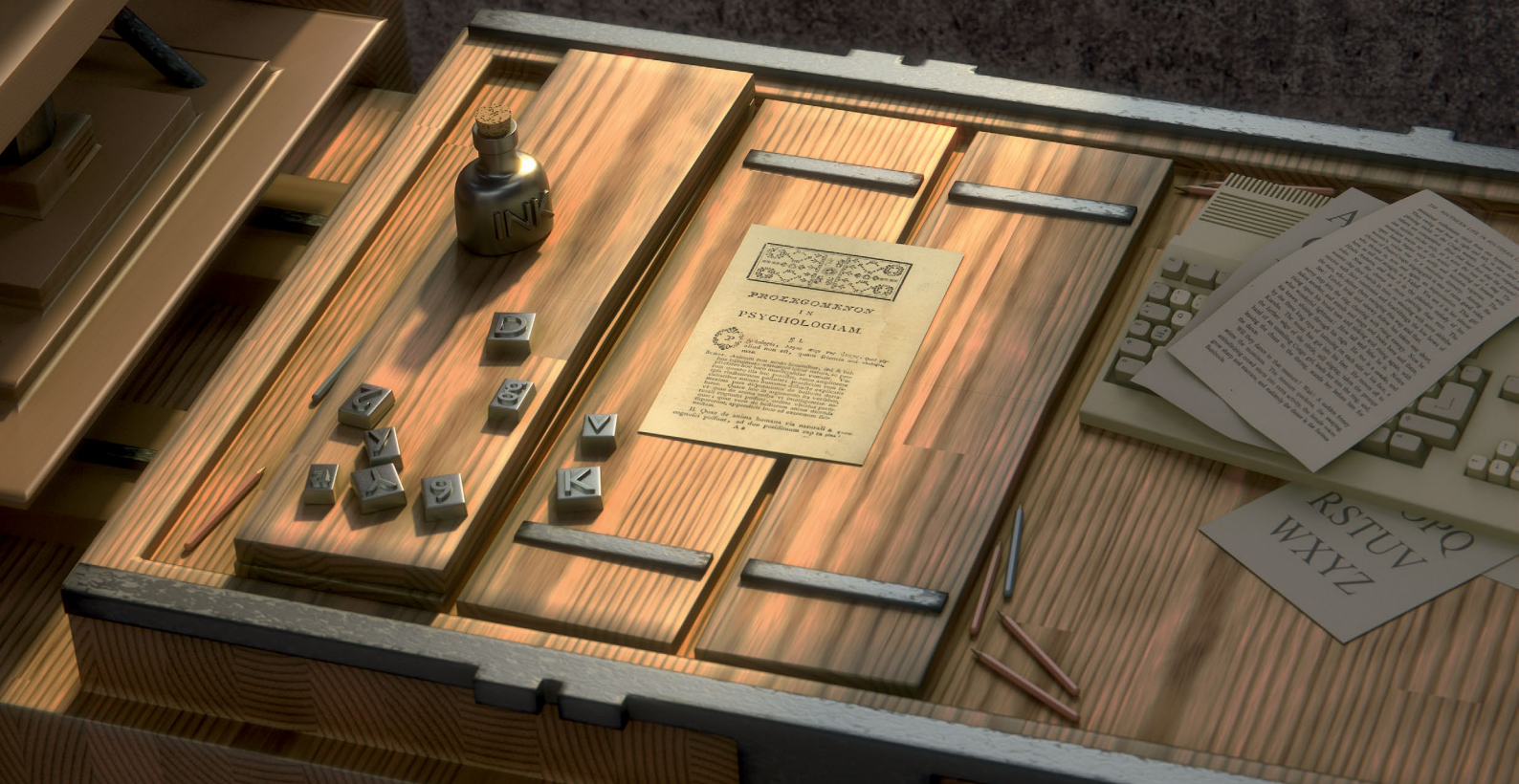
– Nykyään on ihan riittävästi muihin ohjelmointikieliin perustuvia ohjelmointiympäristöjä. Lisäksi kun tarkoituksena on käyttää konekieltä, niin on tietenkin luontevaa kirjoittaa API konekielille. Menuetin API on selkeä ja nopea oppia ja lisäksi helppokäyttöisempi kuin useat muut API:t.

Toivotaan, että harrastelu MenuetOS:n parissa jatkuu ja järjestelmäästä saadaan uusia ja parempia versioita. Alku näyttää jo viehättävältä. MenuetOS:llä voi nyt jo katsella televisiota, irkkailla, surffata webissä, opetella konekieliohjelmointia tai jopa pelata Doomia ja Quakea. Suosittelemme kokeilemaan, jos hyppysissä pysyy ongelmitta joko korppu tai virtuaalikone ja tekee mieli sukeltaa uuteen. 🐉



Skrollin nettisivu ei piirry ihan oikein MenuetOS:n HTTPC-selaimella.

MenuetOS:n sivusto <http://menuetos.net/>
Käyttäjien foorumi <http://menuet.fr.yuku.com/>
IRC-kanava #menuetos @ Ircnet



Tietokone typografian tuhoajana

Tekstin typografia on osa sen viestiä, ja entisajan ammattilaiset hallitsivat sen hyvin. Tietokoneistuminen romahdutti typografian tason monissa viestinnän muodoissa, ja vielä nykyäänkin tekstien ulkoasu heijastelee kirjoituskoneiden ja varhaisten tietokoneiden asettamia rajoituksia.

Teksti: Jukka K. Korpela

Kuvat: Sakari Leppä, Jukka K. Korpela

Jos avaamme minkä tahansa 1900-luvun alkupuolella julkaistun kirjan, saamme käsityksen vanhasta kirjapainotypografiasta. Kun vertaamme sitä nykyaikaiseen tekstinkäsittelyohjelmalla tehtyyn dokumenttiin, näemme jotain paljon alkeellisempaa, karkeampaa – ensimmäisenä huomio kiinnittyy ehkä palstan tasaa-mattomuuteen. Vielä alkeellisempaan päästään tyyppillisessä tietokonetulos-teessa kuten viranomaisen lähettä-mässä kirjeessä, joka on muodoltaan henkilökohtainen mutta ohjelmalla generoitu. Tyyppillisesti se on tuotettu ikään kuin kirjoituskoneen ehdoilla – ja siltä se näyttääkin.

Kuvat 1–3 esittävät typografian kolmea eri tasoa sovellettuina samaan tekstiin: Volter Kilven romaanin *Alastalon salissa* varsinaisen tekstin alkua eri muodoissa: pelkkänä tekstinä Courier New -fontilla (Lönnot-projektin aineistoa), epub-muotoisena e-kirjana (Elisa Kirjan aineistoa) ja ladotusta

kirjasta pdf-muotoon muunnettuna (Suomalaisen klassikkokirjaston aineistoa).

Mitä typografia on?

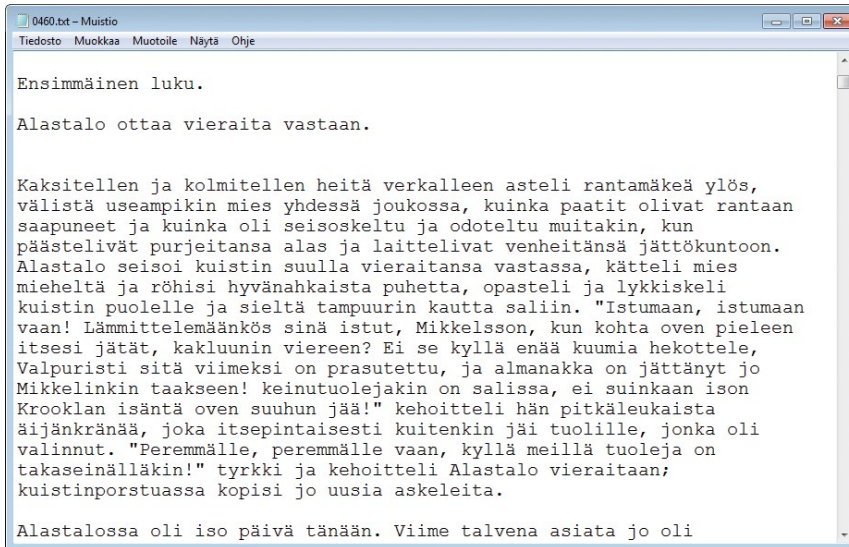
Markus Itkosen *Typografian käsikirjan* mukaan typografia on ”tekstin valmistamista tai sen muotoilua valmiilla – ennakkoon valmistetuilla – kirjaimilla”. Laajemmassa mielessä typografialla voidaan tarkoittaa tekstiä sisältävän julkaisun tai asiakirjan muotoilua muutoinkin.

Aikoinaan ”ennalta valmistettu” tarkoitti irtokirjasimia, joita valettiin metallista sarjatuotantona. Tämä oli ratkaiseva keksintö, joka mahdollisti kirjapainotaidon sellaisena kuin sen tunnettiin. Nykyisin kirjainten esitykset ovat ”ennalta valmistettuja” paljon abstraktimmassa mielessä: fonttitiedosto sisältää ohjeet (algoritmin), jolla ohjelma voi tuottaa kunkin merkin halutunkokoisen esityksen vektorigrafiikalla ja sitten rasteroida sen pikseliesitykseksi.

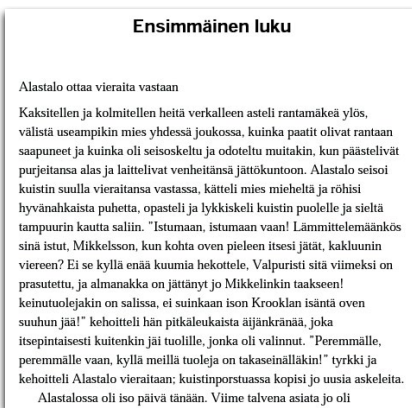
Joka tapauksessa ”ennalta valmistettu” tietyn merkin esitykset ovat keskenään identtisiä. Pois jää käsin kirjoitetun tekstin vaihtelevuus, jossa merkin esitys voi riippua sen asemasta sanassa tai kirjoittajan mielialasta. Luettavuus paranee, etenkin kun käsialojen persoonallinen vaihtelu jää pois. Mutta tätä voidaan pitää myös esteettisenä köyhtymisenä. Kautta aikojen on ollut jännittävä: helppolukuisuus ja kauneus eivät kulje käsi kädessä, tai ainakin tämä riippuu kauneusihanteista.

Miksi typografia on tärkeää?

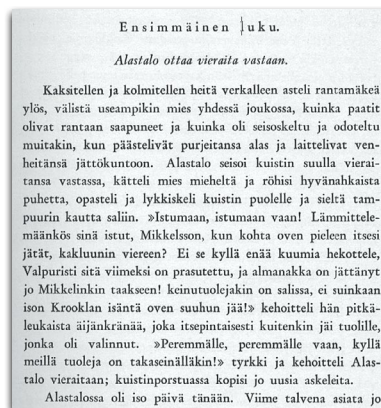
Tekstin muoto on aina viesti, halusimme tai emme. Jos teksti näyttää huolimattomasti ladotulta, sen oletetaan olevan huolimattomasti kirjoitettua, kuten se usein onkin. Eri fontteihin ja muuhun ulkoasuun liittyy mielikuvia: Times New Romanilla ladottu teksti tuo mieleen sanomalehdet, Arial käsikirjat ja käyttöohjeet, Comic Sans ehkä lasten sarjakuvat. Tällaisia asioita on vaikea tutkia tieteellisesti, mutta vai-



Kuva 1. Alastalon salissa -romaanin alku pelkkänä tekstinä.



Kuva 2. Saman romaanin alku e-kirjana.



Kuva 3. Romaanin tekstiä ladottuna vanhanajan menetelmillä.

kutukset ovat silti todellisia.

Tekstin luettavuuteen vaikuttavat suuresti fontti, fonttikoko, tekstin ja taustan värien kontrasti, rivinpituus ja riviväli. Näistä epäselvin on fontin vaikutus, josta on paljon käsityksiä ja myös tutkimuksia mutta tuloksiltaan vaihtelevia. Luettavuuden kannalta onkin olennaisempaa valita muut muotoiluseikat niin, että ne ovat sopivia valitulle fontille.

Ulkoasun miellyttävyyden on vahvasti subjektiivinen ja kiistanalainenkin asia. Julkaistavat tekstit joka tapauksessa yleensä halutaan muotoilla niin, että ne ”näyttävät hyviltä”. Ladotun tekstin esteettinen vaikutelma koetaan usein jopa niin tärkeäksi, että sen nimissä tehdään luettavuuden kannalta selvästi huonoja ratkaisuja.

Typografian ensimmäiset vuosisadat

Kirjapainotaidon keksimisen jälkeen tekstit jakautuivat selvästi kahteen tyyppiin: latojan tekemällä ladelmalla

tuotettuun painotekstiin ja käsin kirjoitettuun tekstiin. Latominen oli ammattityötä, jonka tekijältä vaadittiin sekä kielen osaamista että typografisia taitoja. Se ei ollut pelkkää kirjasimien laittamista järjestykseen, vaan latoja myös huolehti tekstin muotoilusta siisteiksi palstoiksi muun muassa tavutuksen ja välityksen avulla. Välitystä tehtiin lisäämällä kirjasinten väliin kaiteita metallisuikaleita.

Typografiaan syntyi oppeja ja perinteitä, jotka vaikuttavat yhä. Niihin kuuluu käytännöllisiä periaatteita kuten luettavuuden kannalta sopiva rivinpituus, mutta myös esteettisiä valintoja kuten yleensä tekstin tasaaminen molemmista reunoista. Kirjasintyyppiä (fontteja) kehitettiin suuri määrä eri tarkoituksiin, mutta niiden käyttö oli hillittyä: yhdessä julkaisussa käytettiin yleensä vain yhtä tai kahta eri tyyppiä. Vaihtelua saatiin aikaan sekä käyttämällä eri kokoja että kursivilla ja lihavoineilla.

Typografia ja tottumukset

Suuri osa typografiasta on tottumukseen perustuvaa. Entisinä aikoina luettiin vain joko painettuja kirjoja tai käsin kirjoitettuja tekstejä. Kirjapainoalan käytäntöihin totuttiin. Totumus vaikuttaa typografiassakin kahdella tavalla: se luo tuttuutta, turvallisuutta ja helppoutta, mutta toisaalta sitä vastaan kapinoidaan, koska halutaan erottua. Voidaan sanoa, että nykyajan mainonnassa rikotaan kaikkia mahdollisia vanhan typografian sääntöjä, jopa vain rikkomisen takia. Mainoksen on erotettava massasta, muuten sen vaikutus jää liian pieneksi. Tämä johtaa mainosalan sisäiseen kilpailuun ja yhä erikoisempiin ratkaisuihin.

Tietokoneviestintää on harjoitettu niin kauan, että sekin on jo vaikuttanut tottumuksiin. Vaikutus on suurin siellä, missä perinteisesti taitettuja kirjoja ja muita painojulkaisuja luetaan vähiten. On selvää, että esimerkiksi Facebook-kirjoittelussa ei pyritä huoliteltuun asuun sen enempää kielenkäytön kuin typografiakaan osalta. Typografiaan ei juuri ole mahdollisuuksiakaan, sillä Facebookissa ei voi käyttää edes kursivointia eikä muuta fontin vaihtelua. Sama koskee monia muitakin viestinnän muotoja, jotka saattavat muodostaa jopa valtaosan kaikesta, mitä tietokoneajan ihminen lukee.

Olemme laajasti tottuneet tietokoneviestinnän moniin outouksiin niin, että saatamme pitää niitä luonnollisina ja tavanomaisina. Tämä johtaa jopa siihen, että kummeksutaan sellaisia käytäntöjä, jotka ovat kirjapainoalan monisatavuotista ja koeteltua perinnettä. Pieni esimerkki tästä on, että Helsingin Sanomat otti joitakin vuosia sitten käyttöönsä ”vanhanaikaiset” gemenanumerot, jotka ovat korkeudeltaan vaihtelevia (kuva 4). Jokin ajan kuluttua tästä luovuttiin, luultavasti kielteisen palautteen takia. Perinteisen ja perustellun käsityksen mukaan gemenanumerot kuitenkin sopivat tavalliseen ei-tekniiseen tekstiin paremmin kuin taulukoita yms. varten kehitetyt ”nykyaikaiset” numerot.

Se tapahtui 9. elokuuta 1827.

Se tapahtui 9. elokuuta 1827.

Kuva 4. Gemenanumerot (ylempi teksti) olisivat useimmiten sopivampia tavalliseen tekstiin.

Kirjoituskoneiden tulo

Kirjoituskoneet eivät aluksi uhanneet kirjapainotypografiaa, sillä niillä ei tehty painettavia tekstejä vaan kirjoitettiin asiakirjoja. Niitä käytettiin tarkoituksiin, joihin ennen oli käytetty käsin kirjoittamista. Asiakirja tehtiin yhtenä kappaleena tai hiilipaperin avulla kahtena tai muutamana kappaleena. Aikanaan kävi mahdolliseksi monistaa konekirjoitustekstejä vahanomistuksella ja myöhemmin valokopioinnilla. Konekirjoitusjäljestä tuli myös julkaisumuoto.

Konekirjoitusjäljelle on ominaista yhden kiinteän fontin käyttö, ja kyseessä on normaalisti tasalevyinen (monospace) fontti. Tasalevyisyys tarkoittaa, että kaikki merkit vievät saman verran tilaa vaakasuunnassa, kun merkkiin kuuluva tyhjä tila otetaan huomioon. Tämän takia merkkien muodot ovat suorastaan luonnottomia, kun niitä verrataan kirjapainotypografiaan tai käsin kirjoittamiseen. Esimerkiksi i-kirjain on keinotekoisesti venytetty pääteviivojen avulla, ja W- ja M-kirjaimet vaikuttavat sivusuunnassa rutistetuilta.

Tasalevyisyys johtui kirjoituskoneiden mekaniikasta, mutta siitä oli myös etua alkeellisen taulukoinnin toteuttamisessa. Väilyöntien avulla voidaan peräkkäisten rivien kohtia asetella kohdakkain niin, että tulos näyttää taulukolta. Myös tekstin sisennys onnistuu helposti, joskin kömpelösti. Yhä edelleen käytetään ohjelmoinnissa ja ohjelmakoodin esittämisessä yleensä tasalevyisiä fontteja, vaikka nykyisin sisennykset voisi tietysti hoitaa muutenkin.

Konekirjoitustekstistä puuttuvat selkaiset muotoilun peruskeinot kuin *kursiivi* ja *lihavointi*. Värejäkään ei juuri käytetty, vaikka niitä olisi periaatteessa saatu aikaan erivärisillä värinauhoilla. Korostuskeinoiksi jäivät kirjoittaminen **VERSAALILLA** eli suuraakkosilla ja **ALLEVIIVAUS**. Niistä varsinkin versaalin käyttö elää yhä vahvasti, vaikka tarjolla olisi suuri joukko muita keinoja. Alleviivaus jäi laajasti pois käytöstä, kun konekirjoituksesta siirryttiin muihin tapoihin, mutta se tuli takaisin verkkosivujen ja yleensä hypertekstin linkkien esittämisen keinona. Lihavointia ja kursivointia pyritään joskus jäljittelemään tai kuvaamaan erikoismerkkien avulla: *lihavaa* ja /kursiiviat/.

Esitysasun piirre	Kirjoituskonetason esitys	Kirjapainotason esitys
Kappaleen muoto	Vasenuora, liehureuna oikealla, ei tavutusta.	Tasattu molemmista reunoista, tavutus käytössä.
Kappaleiden erottaminen	Tyhjä rivi välissä.	Kappaleen 1. rivin sisennys.
Palstoitus	Yksi palsta.	Tarvittaessa useita palstoja.
Korostus	Versaali tai alleviivaus.	Isompi fontti, kursivi, lihavointi, väri, taustaväri ym.
Otsikkomuotoilu	Tyhjä rivi yllä ja alla. Versaali ainakin ylimmille otsikoille.	Yleensä tyhjää tilaa edellä. Eri muotoilukeinoja kuten iso fontti.
Taulukko	Väilyönneillä "käsin" muotoiltu taulukko.	Ohjelmalla tehty taulukko, johon voi liittyä katsetta ohjaavia viivoja ym.
Fontit	Yksi tasalevyinen fontti, yksi fonttikoko.	Yksi tai useampia fontteja, eri fonttikokoja.
Merkkien muodon vaihtelu	Tietty merkki aina samanmuotoinen.	Merkistä voi olla eri muunnelmia eri yhteyksissä.
Välimerkit	Ascii-merkit " ' -	Perinteiset " ' –
Merkkien välistys	Ei välistystä.	Automaattinen tai manuaalinen välistys.
Kuvitus	Ei mahdollista. Joskus "Ascii-grafiikkaa" tilalla.	Mahdollista.

Taulukko 1. Kirjoituskonetekstin ja ladotun tekstin eroja.

Suppea merkkistö

Kirjoituskoneiden mekaniikasta johtui sekin, että niiden merkkivalikoima oli suppea. Pienestä valikoimasta huolimatta kirjoittaminen vaati niin monia näppäimiä, että merkkivarsien tarttuminen toisiinsa oli todellinen ongelma. Nykyisten tietokonenäppäimistöjen suppeus jatkaa samaa perinnettä, vaikka teknisiä syitä ei enää ole. Perinne elää, koska ihmiset ovat tottuneet näppäinten pienehköön määrään ja nykyisiin näppäinaseteluihin.

Merkkivalikoiman pitämiseksi pieninä kaikki eri kielten lainausmerkit ("«»“„, ym.) korvattiin yhdellä uudella merkillä, pystysuoralla lainausmerkillä ("). Yksinkertaiset lainausmerkit ja heittomerkit (') korvattiin pystysuoralla heittomerkillä ('). Yhdysmerkki (-), ajatusviiva (—), pitkä ajatusviiva (—) ja miinusmerkki (–) korvattiin yhdysmerkiksi sanotulla merkillä (-), jonka oli tarkoitus siis hoitaa neljän eri viivamerkin tehtävää ja muutama muukin.

Vanhoista kirjoituskoneista oli usein jopa numerot 0 ja 1 jätetty pois; niiden tilalla käytettiin al-kirjainta l ja versaali-o:ta O. Vaikka tästä sentään luovuttiin, jäljet saattavat yhä näkyä: monessa fontissa l ja l sekoittuvat toisiinsa, joskus myös 0 ja O. Tavallisessa tekstissä tämä ei haittaa, koodimerkinnoissa sen sijaan ehkä paljonkin. Esimerkiksi kirjoitettaessa lennon varaustunnusta,

jossa on kirjaimia ja numeroita, pitää tietää tai arvata, että O on aina o-kirjain eikä numero 0.

Tietokoneet kirjoituskoneina

Kun tietokoneiden hinta oli laskenut sen verran, että niitä voitiin käyttää tekstinkäsittelyyn, käyttö oli hyvin kirjoituskonetyyppistä. Laitteissa – kuten näyttöpäätteissä ja rivikirjoittimissa – ei juuri ollut mahdollisuuksia muuhun. Ohjelmat olivat tekstieditorin tyyppisiä, ilman muotoilukeinoja. Tiedostomuoto oli tyypillisesti pelkkää tekstiä (plain text).

Osittain elämme vielä tätä aikaa. Nopeaan muistiinpanojen kirjoittamiseen saattaa tekstieditori edelleen olla sopivin. Typografinen ongelma syntyy, kun näin kirjoitettuja tekstejä käytetään asiakirjoina tai jopa julkaistavina teksteinä.

Suomalaiset asiakirjamallit ja -standardit heijastavat nekin vielä suurelta osin kirjoituskoneiden aikakautta. Ne on suunniteltu – usein kai jotenkin itsestäänselvästi – sellaisiksi, että niitä voi käyttää myös kirjoituskonetta käytettäessä. Vasta vuonna 2007 muutettiin yleistä asiakirjastandardia SFS 2487 muun muassa niin, että otsaketietoja ei enää suositella kirjoitettavaksi versaalilla.

Taulukko 1 esittää keskeisiä muotoilupiirteitä, joissa kirjoituskonetekniikka aikoinaan saneli tiettyjä ratkaisuja,

ja vastaavia kirjapainotasoisien tekstin piirteitä. Kuvat 1 ja 3 havainnollistavat näitä eroja. Monet kirjoituskonetasoiset ratkaisut ovat edelleen tavallisia silloinkin, kun käytössä on moderni tekniikka.

Konekirjoitustekstin kaltainen esitystapa on säilynyt monilla aloilla muun muassa siksi, että se toimii kaikilla laitteilla ja kaikissa ohjelmissa, vaikka ulkoasu onkin ruma. Esimerkiksi Internetin tekniset määrittelyt RFC:t julkaistaan yhä pelkkänä tekstinä, joka rakenteeltaan täysin vastaa (Ascii-merkistöön rajoittuvaa) konekirjoitustekstiä. Tosin nykyisin toissijaisena, epävirallisena esitysmuotona on myös yksinkertainen HTML-muoto.

Tekstinkäsittely tuli – ei oltu valmiita

Tekstinkäsittelyohjelmien käyttöönotto muutti asiakirjatuotantoa olennaisesti, mutta typografiaan sen vaikutus oli paljon vähäisempi kuin olisi voinut luulla. Vaikka ohjelmat tarjoavat monipuolisia muotoilumahdollisuuksia, useimmat käyttäjät hyödyntävät niistä vain pientä murto-osaa. Tekstinkäsittelyohjelmilla kirjoitetaan paljolti kuten tekstieditoreilla ja sitä ennen kirjoituskoneilla. Yleensä on sentään opittu olemaan itse rivittämättä tekstiä ja painamaan enter-näppäintä vasta kappaleen lopussa, ja kursivointi ja lihavointi ehkä osataan mutta ei juuri muuta.

Asiakirjojen muodossa vallitsee monia kirjoituskoneiden ajan perinteitä. Kappaleet erotetaan toisistaan tyhjillä riveillä. Tämä on omiaan pilkkomaan esityksen visuaalisesti erillisiksi palasiksi, mikä saattaa sopia tekniseen selostukseen. Painotekstin perinteenä on kappaleen ensimmäisen rivin sisennys uuden kappaleen alun osoittamiseksi; vain laajahkojen tekstijaksojen välissä saattaa olla tyhjä rivi osoittamassa aihepiirin vaihtumista. Muutoinkin tekstinkäsittelyohjelmalla tehdyn dokumentin muoto on usein lähempänä konekirjoitustekstiä kuin painotekstiä. Tosin fontti ei yleensä ole tasalevyinen, koska oletusfontti ei ole sellainen.

Aina, kun tekstien esittämiseen tulee uusia teknisiä mahdollisuuksia, suurin osa ihmisistä käyttää niistä vain pientä osaa, mutta merkittävä vähemmistö innostuu liiaksikin niiden käytöstä. Kun ensin tekstinkäsittelyohjelmat ja

aikanaan myöhemmin verkkosivujen teon ohjelmat tarjosivat mahdollisuuden eri fonttien käyttöön, siinä lyötiin yli raskaasti. Perinteisessä kirjapainoladonnassa käytetään yleensä yhtä fonttia, mutta siitä eri muunnelmia (tavallinen, kursiivi, lihava ja eri koot), tai joskus harkitusti kahta tai kolmeakin fonttia. Mutta liiallinen innostus uusien välineiden käytössä on saanut käyttämään jopa kymmentä eri fonttia samassa dokumentissa.

Jokamiehen julkaisu-oikeus

Tekstinkäsittelyohjelmalla kirjoitettu teksti voidaan kopioida ja tulostaa helposti, ja jopa julkaistavia kirjoja tehdään niin. Itse asiassa jos se tehdään hyvin, tulos on typografisesti parempi kuin osaamaton käyttäjä saa aikaan hyvällä taitto-ohjelmalla. Tekstejä voidaan kirjoittaa myös jaettaviksi verkon kautta tai myytäväksi e-kirjoina.

Sikäli kuin julkaisemisen kynnys on riippunut tekniikasta, se alkoi olennaisesti laskea jo vuosikymmeniä sitten ja on nyt kokonaan poistunut. Julkaisemisen käsitekin on hämärtynt. Ajatellaan yhä, että runokirja on julkaisu mutta yrityksen asiakastiedote ei, vaikka jälkimmäisellä voi olla tuhat kertaa

enemmän lukijoita.

Tämä on toisaalta merkinnyt sitä, että typografia on yhä enemmän aivan amatöörien käsissä. Suurin osa niin sanotusta jokamiehen julkaisemisesta on sellaista, missä typografiaa ei ajatella lainkaan.

Tekniikoiden ja niiden soveltamisen kehitys on johtanut siihen, että julkaisun muotoileminen on siirtynyt erikoistuneesta latojan ammattityöstä jokamiehen työkseen – mutta ilman ammattitaitoa ja riittävää tehtävään paneutumista. Kustantamoissa on kyllä vielä ammattilaisia, mutta hekin hoitavat suurimman osan muotoilemisesta suoraan ohjelmilla, vain hiukan pahimpia ongelmia korjaillen ja parametreja säätäen.

Asiakirjan tai julkaisun muotoilu hoidetaan nykyisin yleensä muun työn ohessa ja nopeasti. Paljon pelastaisi se, että siinä käytettäisiin hyviä dokumenttipohjia, joissa perusmuotoilu on asetettu hyvin. Niin ei kuitenkaan yleensä tehdä. Pohjaksi saatetaan kyllä ottaa jokin vanha dokumentti, mutta harvemmin sellainen, jota kannattaisi käyttää. Niinpä tuotteissa on tyyppillisesti useita typografisia perusvirheitä (ks. oheinen tietolaatikko).

Typografisia perusvirheitä

- Liian pitkä rivinpituus. Typografinen suositus on noin 55–60 merkkiä rivillä, mutta ei ole harvinaista, että pituus on yli 100 merkkiä, jolloin luettavuus kärsii selvästi.
- Liian pieni riviväli. Tämä on kytköksissä edelliseen, koska rivivälin pitäisi olla sitä suurempi, mitä suurempi rivinpituus on.
- Tavutuksen puute silloinkin, kun se aiheuttaa pahaa jälkeä ja automaattinen tavutus olisi helposti otettavissa käyttöön.
- Tavutusvirheet. Jos automaattinen tavutus on käytössä, tulee vääriä tai huonoja tavutuksia (kuten kansa-nedustaja) noin yksi kappale sivua kohti, ellei tulosta tarkisteta.
- Liian pieni fontti. Asiakirjastandardin suositaman 10–12 pisteen sijasta joudutaan usein tihrustamaan selvästi pienempää tekstiä. Ongelman vakavuus riippuu muun muassa fontista ja siitä, onko kyseessä painotuote vai ruudulta luettava teksti.
- Tausta sotkee. Etenkin lehdissä on tavallista, että osalla tekstistä on taustanaan kuva tai värillinen alue. Tähän virhetyyppiin voidaan lukea myös etenkin verkkosivuilla tavallinen tapa käyttää kyllä valkoista taustaa mutta tekstin värinä harmaata niin, että tekstin ja taustan kontrasti on riittämätön.
- Otsikot eivät erotu. Otsikot on saatettu kirjoittaa jopa aivan samalla tyyllillä kuin tekstikappaleet.
- Liikaa fontteja. Vaikutelma voi olla erittäin sekava, ja lukija joutuu arvaamaan, mitä on tarkoitettu korostaa, kun tekstissä on toinen toistaan erilaisempia fontteja.
- Väärä fontti. Esimerkiksi kokeileva, erikoinen fontti leipätekstissä on lähes aina selvä virhe, samoin Comic Sans asialliseksi tarkoitettussa tekstissä.
- Merkkitaso virheet kuten yhdysmerkki "–" ajatusviivan "—" tai miinusmerkin "–" tilalla tai "pystyt" lainausmerkit "kaarevien" tilalla.

Tulosteiden typografia

Nykyaikaisessa hallinnossa ja elinkeinoelämässä tuotetaan suuria määriä dokumentteja, jotka luodaan tietokannasta tai muuten automaattisesti. Usein ne ovat ”henkilökohtaisia” kirjoituksia, joissa yleiseen kirjepohjaan on ohjelmallisesti sijoitettu tietyn vastaanottajan tiedot. Ehkä on hyväkin, että sellaiset viestit näyttävät persoonattomilta, tietokonemaisilta. Joka tapauksessa yleensä olisi mahdollista tuottaa paljon parempaakin jälkeä.

Generoidut dokumentit ovat yleisesti ”konekirjoitusmuotoisia” jopa niin, että fontti on tasalevyinen (kuva 5). Sellaisen tuottaminen voi olla teknisesti helpompaa, mutta paljolti kyse on vain tavasta ja vanhojen ohjelmien käytöstä.

Nykyisin tuotetaan myös laadukkaampaa jälkeä, mutta silloinkin typografisessa muotoilussa tehdään yleisesti virheitä. Etenkin liian pitkät rivit ja liian pieni fontti ovat tavallisia.

Verkojulkaiseminen ja e-kirjat

Kun verkkojulkaiseminen alkoi 1990-luvulla, se oli aluksi lähinnä kirjoituskonemuotoista sekä Gopher-järjestelmässä että sen melko pian syrjäyttäneessä WWW:ssä. Jo muutama vuoden kuluttua tuli kuitenkin mahdollisuus käyttää eri fontteja, eri fonttikokoja, lihavoitteen ja muita korostuksia sekä myös kuvia. Tilanne muodostui siis suunnilleen samaksi kuin varhaisessa tekstinkäsittelyssä.

Sen jälkeen WWW-julkaisemisen tekniikat ovat typografian kannalta kehittyneet hitaasti. Suuri osa typografisesti merkittäviä tekniikoita on nyt toteutettu vain osaan nettiselaimista ja osittain, joten niitä ei vielä juuri käytetä. Esimerkiksi tasalevyinen palsta on melko harvinainen, ja oikealla tavalla – tavutusta käyttäen – tehty tasapalsta on vielä paljon harvinaisempi.

Historia on toistanut itseään e-kirjojen tekniikoissa. Vaikka e-kirjaformaattien määrittelyt sallivat modernin HTML:n ja CSS:n, todellisuudessa se, mikä toimii useimmissa lukuohjelmissa, on suunnilleen samalla tasolla kuin verkkojulkaiseminen noin vuonna 1995. Monet lukuohjelmat jättävät lähes kaikki e-kirjan tekijän tekemät muotoilut huomiotta ja enimmäkseen vain näyttävät tekstisisällön omalla tavallaan, fonttia myöten.

Hei. Työnhakusi on uusittu ja on voimassa toistaiseksi. Käytännössä tämä tarkoittaa, ettei sinun tarvitse itse uusita työnhakusi jatkossa. TE -toimisto uusii työnhakusi automaattisesti n. 6 kk:n kuluttua, mikäli et ole ilmoittanut muutoksesta ennen sitä. Työnhakuun liittyvät muutokset voit itse suorittaa verkkopalvelussamme osoitteessa www.te-palvelut.fi - Asioi verkossa Oma Asiointi Henkilöasiakkaalle. Yhteydenottoihin voit käyttää puhelinnumeroa: 0295 040 000 tai sähköpostiosoitetta: kirjaamo.uusimaa@te-toimisto.fi ystävällisin terveisin

Kuva 5. Viranomaisen kirjeen typografiaa vuonna 2015.

Historian raskas painolasti

Tavallinen tietokoneen näppäimistö on yhä olennaisesti samanlainen kuin 1800-luvun mekaanisissa kirjoituskoneissa ja 1950-luvun tietokoneissa. Myös kosketusnäyttölaitteissa on yleisesti pyritty qwerty-näppäimistöön sen tuttuuden takia. Emme tunnu pääsevän irti näppäimistön yleisrakenteesta, vaikka sen suunnittelua ohjanneet mekaanisten vipuvarsien liikkeet ovat jo historiaa. Seuraukset näkyvät typografiassa merkkien valinnan tasolla. Harva käyttää päivittäisessä kirjoittamisessaan sellaisia merkkejä, joita ei löydy näppäinhatuista.

Kirjoituskoneet ovat muutenkin vaikuttaneet tekstien ulkoasuun vahvasti, muun muassa luomalla liehureunaisen, tavuttamattoman ja usein liian leveän palstan. Myös sellainen vaikutus pysyy hyvin sitkeästi. Harva käyttää enää kirjoituskonetta, mutta monet käyttävät tietokonetta miltei kuin kirjoituskonetta ja tuottavat sen mukaista jälkeä. Esimerkiksi tekstinkäsittelyohjelman tyylien käyttö on jäänyt jonkin tietokonekurssin harjoituksiin, jos mitään kurssia on käytykään. Tulokset ovat tyyliittömiä.

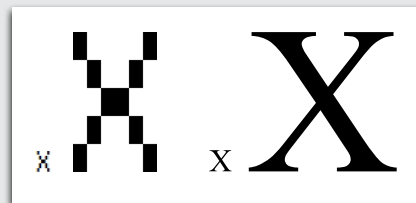
Tyyliittömyyden maailmassa pienikin pyrkimys tyyliin erottaa massasta. Jos joku vaikkapa kirjoittaa pöytäkirjan tai muistion niin, että edes alkeellisia typografian ja hyvän asiakirjakäytännön sääntöjä noudatetaan, niin sehän näyttää melkein painotekstillä. Lukija alkaa helposti ajatella, että onpa tämä tehty huolellisesti. Riskinä toki on, että laatua pidetään snobbailuna. Riski kannattanee ottaa, sillä nykyaikaisessa viestinnässä – oli se esimerkiksi markkinointiviestintää tai tieteellisten tulosten julkaisemista – on yhä tärkeämpää erottua massasta. 🦋

Kirjasimista fonttiedostoihin

Artikkelin alussa siteeratussa typografian määritelmässä käytetään ilmausta ”valmiilla – ennakoon valmistetuilla – kirjaimilla”. Kirjaimella tarkoitetaan tällöin minkä tahansa kirjoitusmerkin erityistä ulkoasua, glyyfiä. Aluksi kirjaimet tässä mielessä olivat fyysisesti ennakoon valmistettuja, metallista valettujen kappaleiden, kirjakeiden, pinnassa olevia kohokuvia, kirjajimmiä, jotka löivät jäljen paperiin.

Nykyisin ”ennalta valmistettuja” ovat merkkien muodot, jotka sisältyvät fonttiedostoihin. Fontti voidaan tulkita tietokoneohjelmaksi, joka sisältää eri glyyfejä piirtäviä aliohjelmia. Vanhoissa rasterifontteissa kukin glyyfi esitettiin kuvan tavoin bittikarttana, mutta nykyaikaisissa vektorifontteissa glyyfin tuottaminen perustuu vektorigrafiikkaan. Tällä on olennainen merkitys tuotettaessa eri fonttikokoja, sillä vektorigrafiikka on täysin skaalautuvaa: glyyfi voidaan suurentaa tai pienentää tarkkuuden kärsimättä – joskin näyttölaitteen esitystarkkuus eli resoluutio vaikuttaa lopputulokseen.

Varhaiset tietokonefontit olivat rasterifontteja. Jopa monille tutuksi tullut Courier on nykyisin yleensä korvattu vektorifontilla Courier New. Toisaalta yhä on käytössä muun muassa GNU Unifont, joka on merkkivalikoimaltaan hyvin laaja mutta ulkoasultaan karkea. Kuva 6 havainnollistaa, miten Unifontin suurentaminen merkitsee karkean pikseliesityksen pikselien kasvattamista, kun taas vektorifontissa (tässä Times New Roman) muoto pysyy normaalina.



Kuva 6. Rasterifontin suurentaminen aiheuttaa usein rakeisuutta, vektorifontin ei.



Päivitimme laitteesi, se on nyt pilalla!

Kuka rakastaisi vanhaa androidia?

Mikko Heinonen

Pidän makuuhuoneen pöydällä Android-tablettia, jolta katselen satunnaisia nettisisältöjä ennen nukahtamista. Tässä tarkoituksessa on palvellut reilun parin vuoden ajan Google Nexus 7 (2012).

Tarvitsemani ohjelmat toimivatkin siinä oikein mukavasti, kunnes Google työnsi laitteeseen Android 5 -päivityksen. Heti käynnistyksen jälkeen kävi ilmi, että tablettiparka on läikähtäjä jo pelkän käyttöjärjestelmän alla. Siirtyminen sovellusten välillä on tahmeaa, ja etenkin videoita hidastui selvästi.

Kun mainitsin asiasta pitkän linjan Android-käyttäjille, sain vakiovastauksen: kikkeliskokkelis, mitäs päivitit. Vanhan laitteen omistajan olisi kannattanut pitäytyä vanhassa versiossa, joka kuluttaa vähemmän resursseja. Tämä on ongelmallista siksi, että Androidista on löydetty myös tietoturvaluutteita ja samaista tablettia on tullut käytettyä myös reissun päällä vaikkapa lippujen varaamiseen. Haluan laitteen olevan tietoturvan suhteen niin ajan tasalla kuin mahdollista, ja näitä päivityksiä on luvassa vain uusimpaan versioon.

Joku sentään välittää

Samalla tajuan olevani onnellisessa asemassa. Googlea kiinnostaa kääriä uusia ohjelmistoversioita näihin referenssilaitteisiinsa peräti kahden vuoden ajan niiden julkaisusta. Taloudesta löytyy myös uudempi kiinalainen halpatabletti, jonka käyttöjärjestelmäpäivitysten määrä viimeisen vuoden ajalta on pyöreä nolla. Sen kanssa ollaan jatkossa foorumeiden kotipolitoisten zip-pakettien varassa.

Eikä orvoksi jääminen rajoitu marginaalivalmistajiin. Takuukorjatut, hal-

vat Iphone 4:t aiheuttivat viime vuonna ostoryntäyksiä myymälässä jos toisessakin. Harva taskurahahintaista muotiluuria hamunnut tuli ajatelleeksi, että juuri hankitussa puhelimessa on ainakin yksi vakava tietoturva-aukko, jota Applella ei ole pienintäkään aikomusta paikata. Viisi vuotta vanha laite on end-of-life-tuote, jolle ei enää tehdä uusia käyttöjärjestelmäversioita.

Asennemuutosta ilmassa

Windows XP:n tuen loppuminen oli sekin kuuma puheenaihe viime vuonna. Vanhalta sotaratsulta vedettiin lopulta piuhat irti 13 vuoden iässä, ja moni jäi sitä edelleen kaipaamaan. Maailmalla on edelleen miljoonittain XP-asennuksia, joista koitunee jatkossa monenlaista riemua niin koneiden omistajille kuin muun muassa tietoverkkojen ylläpitäjille.

XP:n päivittäminen tuli Microsoftille kalliiksi. Hankkimalla halvan OEM-lisenssin vuonna 2001 sai nauttia ilmaisista päivityksistä yli vuosikymmenen ajan. Uudet versiot eivät tarjonneet mitään ratkaisevan erilaista, joten XP oli liian pitkään liian suosittu, jotta sen tuen olisi voinut vain katkaista. Myös Windows 7:n tukea jouduttiin jatkamaan vuoteen 2020 asti, koska siirtyminen XP:stä siihen alkoi jälkijunassa.

Tämän ongelman Microsoft pyrkii selvästi välttämään Windows 10:n kanssa. Ilmaiseksi tarjottavalla päivityksellä yritetään hivuttaa käyttäjät uusimpaan versioon ja keventää historian painolastia. Sivu on lainattu kilpailijan opikirjasta, sillä Apple on jakanut OS X -päivityksiä ilmaiseksi jo pitempään. Erona on tietenkin se, että OS X toimii (ilman kikkailua) vain Applen laitteistolla. Microsoft joutuu subventoimaan ilmaisipäivitystä mainostuloilla.

Omenatkin happanevat

Sukulaisnainen kyseli jokunen kuu-

kanssa. Kyseessä on ehkä 6–7 vuotta vanha läppäri, joka toimii vielä aivan hyvin ja riittää käyttötarkoitukseensa. Valitettavasti vain sen näytönohjain on valmistajan mielestä niin vanha, ettei koneeseen enää saa uutta käyttöjärjestelmää. Tämä taas aiheuttaa sen, ettei monikaan uudempi ohjelmisto edes asennu. Totesin, että modernimman softan toivossa on joko siirryttävä avoimen lähdekoodin vaihtoehtoihin tai suunnattava myymälään.

Toinen tuttava taas onnistui kyllä päivittämään vanhan Macinsa, mutta tuloksena oli suorituskyvyn täydellinen romahtaminen. Muutaman viikon kuluttua hän päätti lopettaa kärvistelyn ja ostaa uuden tietokoneen. Kauppaan suuntasin muuten minäkin tuskastut- tuani oman tablettini hidasteluun.

Kävelevät aikapommit

Tarjoamalla uusimman päivityksen ilmaiseksi valmistajat itse asiassa karsivat omia tehtäviään. Kun antiikkisetkin raudat vedetään perässä uusimpaan koodipohjaan, ei tarvitse murehtia edellisten julkaisujen korjaamisesta. Ajoittain sitten vain päätetään, mitkä sukupolvet tiputetaan keltasta.

Vanhan laitteen omistajalle ajatus uudesta käyttöjärjestelmästä on hieno siihen asti, kunnes hän tajuaa, että kyseessä on ehkä valinta ruton ja koleran väliltä. Hyväksynkö tietoturvan puutteet ja muut ongelmat vai hidastanko laitteen sietokyvyn rajamaille?

Päätös on helppo niin kauan kuin kuluttajalla on varaa ja tahtoa uusia laitteistoaan parin vuoden välein. Näin ei kuitenkaan läheskään aina ole, ja ne vanhatkin laitteet päätyvät usein jollekin käyttöön. Maailmalla on kasapäin epäkuranttia rautaa, jonka ohjelmistoja kukaan ei enää päivitä. Keskimääräinen käyttäjä ei ymmärrä, miksi toimiva laite pitäisi vaihtaa uuteen, jos sillä saa tehtyä tarpeelliset asiat. Tai miksi pitäisi siirtyä GNU/Linuxiin viimeistään silloin, kun laitteistoon alkaa kertyä vuosirenkaita.

Onkin totta, että Windows XP:llä saa avattua netistä uutissivut. Vanhalla Androidilla tai iOS:llä varustettu puhelin käy edelleen soittamiseen ja facebookailuun. Ongelmaksi tämä muodostuu siinä vaiheessa, kun näitä eläviä kuolleita on miljoonittain kytkettyä maailmanlaajuiseen tietoverkkoon. Riittävän herkullisen reiän löydyttyä niistä syntyy kokonainen zombiarmeija. 🧟



Teekkareiden ja taideopiskelijoiden yhteinen olohuone

Nintendonyyttarit, Darrabingo, LOVE-69 Deittailutekniikan perusteet, TEDxOtaniemi, open mic -ilta sekä VJ-työpaja. Nämä ja monet muut tapahtumat hellivät eri alojen opiskelijoita ja kaupunkilaisia Otaniemi pop up -olohuoneessa toukokuussa lähes kolmen viikon ajan.

Teksti: Elli Kurkikangas

Kuvat: Antti Kurkinen, Essi Laurila

Olohuoneeseen eli Otaniemi Pop Up Living Roomiin sai tulla kuka tahansa, eikä pelkästään osallistumaan tapahtumiin, vaan myös kokeilemaan tapahtuman järjestämistä. Matalan kynnyksen kohtaamispaikka oli kaikille avoin, mutta kohderyhmänä olivat erityisesti otaniemeläiset opiskelijat. Tapahtumia sai järjestää ilman aiempaa kokemusta.

Olohuoneen tapahtumat eivät rajoituneet pelkästään läsnä olleisiin, vaan joitakin niistä saattoi myös seurata etänä nettistriimin kautta.

Otaniemessä opiskelee teekkareiden lisäksi nykyään enenevässä määrin Arabian kampukselta siirtyneitä taide-, media- ja arkkitehtuuriopiskelijoita sekä jatkossa myös kaupparakenteakoulun opiskelijoita. Eri opiskelijaryhmien välinen kohtaaminen ei tapahdu itses-

tään. Yksi Pop up -olohuoneen idea olikin synnyttää AaltoArts-kulttuuria eli entisen Taideteollisen korkeakoulun opiskelijoiden kulttuuria myös Otaniemen kampukselle.

– Paikan tarkoitus oli olla kokeilemisen tila. Otaniemessä on paljon erilaisia kerhoja ja järjestöjä. Tavoitteena oli, että eri opiskelualojen ja järjestöjen ihmiset tutustuisivat toisiinsa yhdessä tekemisen kautta, idean alullepanija Essi Laurila kertoo.

Maanalainen studio

Olohuone oli Otaniemi Underground Broadcasting Systemin, nykyään OUBS ry:n tiloissa, jossa on tv-studio-otason varustus. OUBS ry:n puheenjohtaja Antti Kurkinen suunnitteli ja vastasi Olohuoneen tapahtumien teknisestä toteutuksesta. Hän huolehti siitä, että mikrofonit, mikserit, valot, johdot, sähköt ja muu laitteisto oli kyt-

etty oikein sekä siitä, että joku osaava käyttäjä oli paikalla.

– Olohuone oli täydellinen testi OUBS:n studioille. Studioissa on aiemminkin järjestetty tapahtumia. Kuitenkin erityisesti kesän remonttia ja syksyn vuokratyöitä silmällä pitäen oli hyvä saada kokemuksia erilaisten tilaisuuksien järjestämisestä. Tapahtumien tarpeiden mukaan osasimme paremmin suunnitella rempattavia asioita, kuten ääni- ja kuvakaapeliveitoja sekä tavaroiden säilytysratkaisuja, Kurkinen kertoo.

OUBS:n laitteisto, kuten valot, tykit, mikrofonit ja monikameratuotanto järjestettiin tapahtumakohtaisesti Olohuoneen käyttäjien tarpeisiin. Tekniikkavaatimukset olivat varsin erilaisia riippuen siitä, oliko kyseessä bändin keikka, livelähetys vai työpaja. Esimerkiksi Äänenkäyttö shamanismissa -päivä kuvattiin monikamerateknii-

kalla ja editoitiin myöhempää käyttöä varten. Myös editointi tapahtui OUBSin resursseilla.

- Suurin osa tapahtumista oli teknisesti helppo järjestää ja vaati lähinnä taustamusiikkia, videotykin ja valot päälle. Suurimmat haasteet OUBSille heittivät Äänenkäyttö shamanismissa -päivän äänihoitajan taltiointi sekä Skrollin livestream. Kummatkin taltiointiin usealla kameralla ja mikrofonilla ja kestivät useamman tunnin, Kurkinen toteaa.

- Olohuone oli monipuolinen projekti, jota oli mielenkiintoista tehdä. Päivittäiset tilaisuudet työllistivät paljon, mutta niistä saatu hyvä palaute antoi voimaa.

Matalan kynnyksen hyppy tuntemattomaan

Aalto-yliopiston teknillisen fysiikan ja matematiikan opiskelija Lauri Nieminen osallistui aktiivisesti Olohuoneen tapahtumiin.

- Yleensä Otaniemessä on ainejärjestön juhlia ja perusjuttuja. Pop up -olohuoneessa oli tästä poiketen monipuolisia tapahtumia ja itsensä kehittämiseen liittyviä juttuja. Olohuone oli hyppy tuntemattomaan matalalla kynnyksellä. Jos tuli yhtenä päivänä tapahtumaan, saattoi seuraavana päivänä olla taas jotain ihan uutta, Nieminen kertoo kokemuksistaan.

Hän piti kiinnostavimpana Äänenkäyttö shamanismissa ja nykypäivänä -työpajaa. Allegra Lab Helsinki ja äänihoitaja Merja Valve järjestivät yhteistyönä harjoituksia ja teoriaa yhdistävän työpajan äänen käytöstä ja sen erilaisista kulttuurisista merkityksistä. Lisäksi keskusteltiin tekniikoiden shamanisesta luonteesta, äänen resonanssista ja sen terveysvaikutuksista.

- Sai rentoutua ja osallistua äänihoitoon, muttei tarvinnut aktiivisesti toimia. Kaikki olivat yhdessä mukana kokemuksessa, Nieminen kuvailee.

Otaniemi pop up -olohuone oli onnistunut kokeilu, vaikkakin kävijämäärä vaihteli paljon eri tilaisuuksissa. Yhteensä kävijöitä oli muutama sata. Määrä tosin tuplaantuu, kun mukaan lasketaan Skrollin Tietokonejournalismia ja supervoimia -keskustelutilaisuuteen etänä osallistuneet. Tilaisuudesta uutisoitiin Skrollin 2015.2-numerossa.

Essi Laurila toivoo, että jatkossakin tiloja voisi osan ajasta käyttää saman-



Otaniemi Pop-up Olohuone oli matalan kynnyksen kohtaamispaikka Otaniemessä, jossa jokainen halukas sai toteuttaa ideoitaan, järjestää tapahtumia ja työpajoja tai tuoda luovuuttaan esille.



Darrabingon finaalissa kilpailtiin Alepan lahjakortin pääpalkinnosta. Bingoisäntänä oli Jarno Kokkonen.

tyyppiseen opiskelijatoimintaan kuin toukokuussa. Myös Lauri Niemisen mukaan jatko olisi toivottavaa.

- Aivan ehdottomasti! Otaniemessä on Suomen nerokkaimmat ihmiset ja Olohuone antoi Artsin toteuttamana uusia vaihtoehtoja teekkareille. Humaanille kosketukselle on tarvetta, hän sanoo.

- Olohuone voisi toimia esimerkiksi kerran viikossa OUBSin kertavuokratavan tilan lisäksi. Tähän tarvittaisiin sopiva porukka tekijöitä, Nieminen jatkaa.

Antti Kurkisen mukaan tila toimii hyvin erilaisten tapahtumien järjestämisessä. OUBS ry on mielellään mukana jatkossakin erityisesti tekniikkaa ja kuvaamista vaativissa projekteissa. Tällaisia projekteja voivat olla esimer-

kiksi Skrollin livestreamin kaltaiset tilaisuudet, bändi-illat, äänitykset ja kuvaukset.

OUBS ry vuokraa studiota Aalto Yliopiston Ylioppilaskunnan jäsenille sekä muille kiinnostuneille AYY:n kertavuokrattavana tilana. Perushinta sisältää videotykin ja perustason äänentoiston. Lisähintaan saa käyttöönsä paremman äänentoiston sekä studion katossa olevat valot. OUBS ry:n kanssa sopimalla voidaan toteuttaa muutoksia valaistukseen tai sisustukseen, sekä mahdollistaa livemusiikin toisto tai livekuvan lähetyksen studiossa. Tarkemmat tiedot vuokraamisesta löytyvät syksyllä osoitteesta oubs.ayy.fi.

Tuottajan näkökulma: unelmasta yhteisöllisestä olohuoneesta

Teksti: Essi Laurila

Otaniemi pop up -olohuoneen toukokuisten päättäjäisten jälkeisenä aamuna heräsin studion sohvalta yksin, nauraen raikuvasti ääneen. Olin toteuttanut unelmani ja olin siitä onnellinen. Edellisenä iltana olin pyörinyt innoissani kuin hyrrä ja halunnut tanssittaa OUBS ry:n tekniikan pojat läpi kiitollisuudesta heidän työilleen. Ilman heitä projekti ei olisi onnistunut.

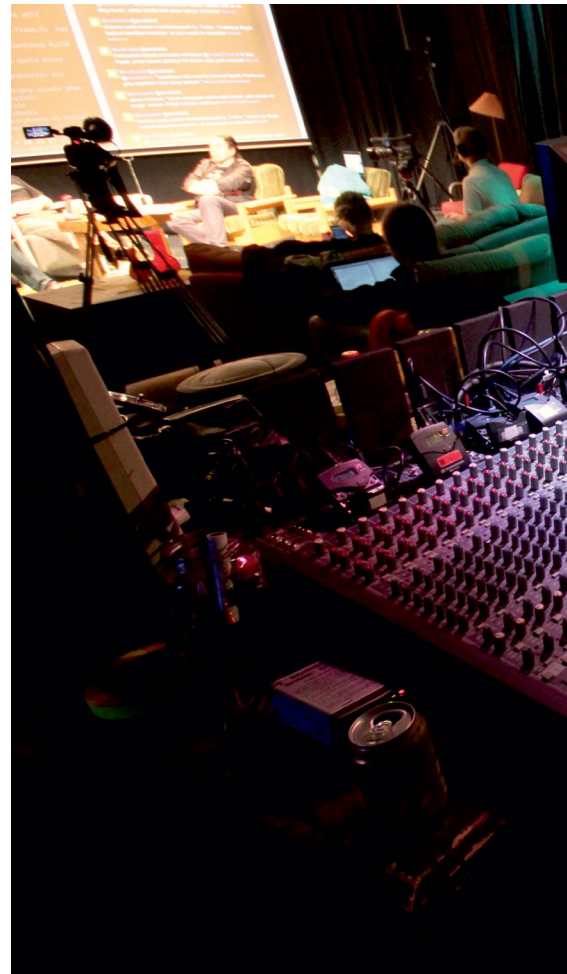
Pop up -olohuone syntyi tarpeesta luoda yhteisöllinen tila, jossa opiskelijat voisivat tavata toisiaan koulutusalaan, teekkarikerhoon tai sukupuoleen katsomatta. Otaniemi oli tähän herkullinen paikka, koska se on enenevissä määrin eri koulutusalojen opiskelijoiden sulatusuuni. Otaniemessä on paljon eri järjestöjen tiloja, muttei tietääkseni yhtäkään kaikille avointa, yhteisöllistä ja opiskelijalähtöisesti koordinoitua tilaa, jonne kuka tahansa voisi tulla osallistumaan ja käyttämään luovuuttaan. Pop up -olohuoneessa voisi myös järjestää tapahtumia ja työpajoja matalalla kynnyksellä ja ilman kustannuksia.

Inspiraationi yhteisöllisiä tiloja kohtaan syntyi jo pari vuotta sitten Tampereella. Paikallinen kaveriporukka vuokrasi Pispalasta liikehuoneiston, Mahdollisuuksien tilan, jonka toiminta perustui yhdessä tekemiselle ja leikille. Meininki oli luovuuteen kannustavaa, ja uskalsin järjestää siellä ensimmäiset Leikkisät konemusabileet -nimiset kokeelliset juhlat. Olin todella inspiroitunut hyvin menneestä kokemuksesta ja muutenkin paikan hengestä, mutta päätin muuttaa järkisyyttä toiseen kaupunkiin. Minua harmitti, että menetin mahdollisuuden ja unelman toimia ympäristössä, jossa ihmiset inspiroivat toisiaan. Siten tajusin, että vastaavanlaiselle tilalle oli tarvetta muuallakin.

Hetkeä myöhemmin löysin itseni Otaniemestä opiskelemasta taiteen maisteriksi. Viime maaliskuussa jutelin kahden opiskelutoverini, Laura Mellasen ja Emilia Hyvösen, kanssa kahvipöydän ääressä ja tajusimme, että meillä oli samanlainen visio matalan kynnyksen tilasta, joka olisi kaikille avoin.

Tuumasta toimeen ryhdyttyämme saimme TTER-rahastolta ja Tokyory:ltä yhteensä vajaat 2000 euroa apurahaa. OUBS ry tarjosi pitkään varastona ollutta AYY:n omistamaa studio-tilaansa kolmeksi viikoksi projektille remonttiapua vastaan. Vapusta eteenpäin tiloissa järjestettiin taideopiskelijoiden ja teekkarien yhdessä pyörittämä Otaniemi pop up -olohuone. Kuka tahansa sai buukata Olohuoneen kalenteriin haluamansa ajan erilaisten tempausten järjestämiselle, ja pian kalenteri alkoikin täyttyä.

Olohuone oli projektina innostava kokeilu, mutta sen suosio tuotti yllättävän paljon työtä. Olin OUBS ry:n puheenjohtajan Antti Kurkisen kanssa



Pop up -olohuoneen tapahtumien taustalla oli paljon teknistä valmistelua ja operointia. Kuvassa Harri Häivälä ja Lauri Hälinen miksauspöydän ja kuvamikserin äärellä.



Skrollin tapahtuma kuvattiin monikameratuotantoa käyttäen. Kameran takana Jukka Kiiski.



paikalla useampina päivinä ja pistäytyivät studiolla ohikulkumatkallaan. Yllättävin olohuonevieras oli dubailainen elämäntaitovalmentaja Hassan al Natour, joka sattui paikalle eräänä iltana ja intoutui itsekkin pitämään it-seluottamusta käsittelevän Ultimate Confidence -työpajan.

Henkilökohtaisella tasolla Pop up -olohuone oli huikea kokemus, jolle toivoisin jatkoa tulevaisuudessakin. Kävijöiden palaute projektista loi uskoa tulevaan ja osoitti, että olohuoneen tapaisille paikoille on Otaniemessä kysyntää. OUBS ry:n studio on syksystä lähtien Aalto-yliopiston ylioppilaskunnan kertavuokrattava tila, joten sen funktio hieman muuttuu. Vuokrattavanakin tila tarvitsee näkyvyyttä, käyttöä ja sisältöä, joten uskon, että projektilla on potentiaalia myös jatkossa. Tavoitteena olisi koota sopiva porukka eri alojen opiskelijoita, joiden kanssa voisi jatkaa Olohuoneen koordinoitua ja OUBS ry:n kanssa yhteistyön tekemistä. 🐼

lähes joka päivä studiolla pitkää päivää. Antti valmisti tekniikan ja minä tuotin, tiedotin sekä hostasin tapahtumia. Lisäksi OUBS ry:n muut jäsenet auttoivat kuvaamisessa ja muissa teknisissä hommissa.

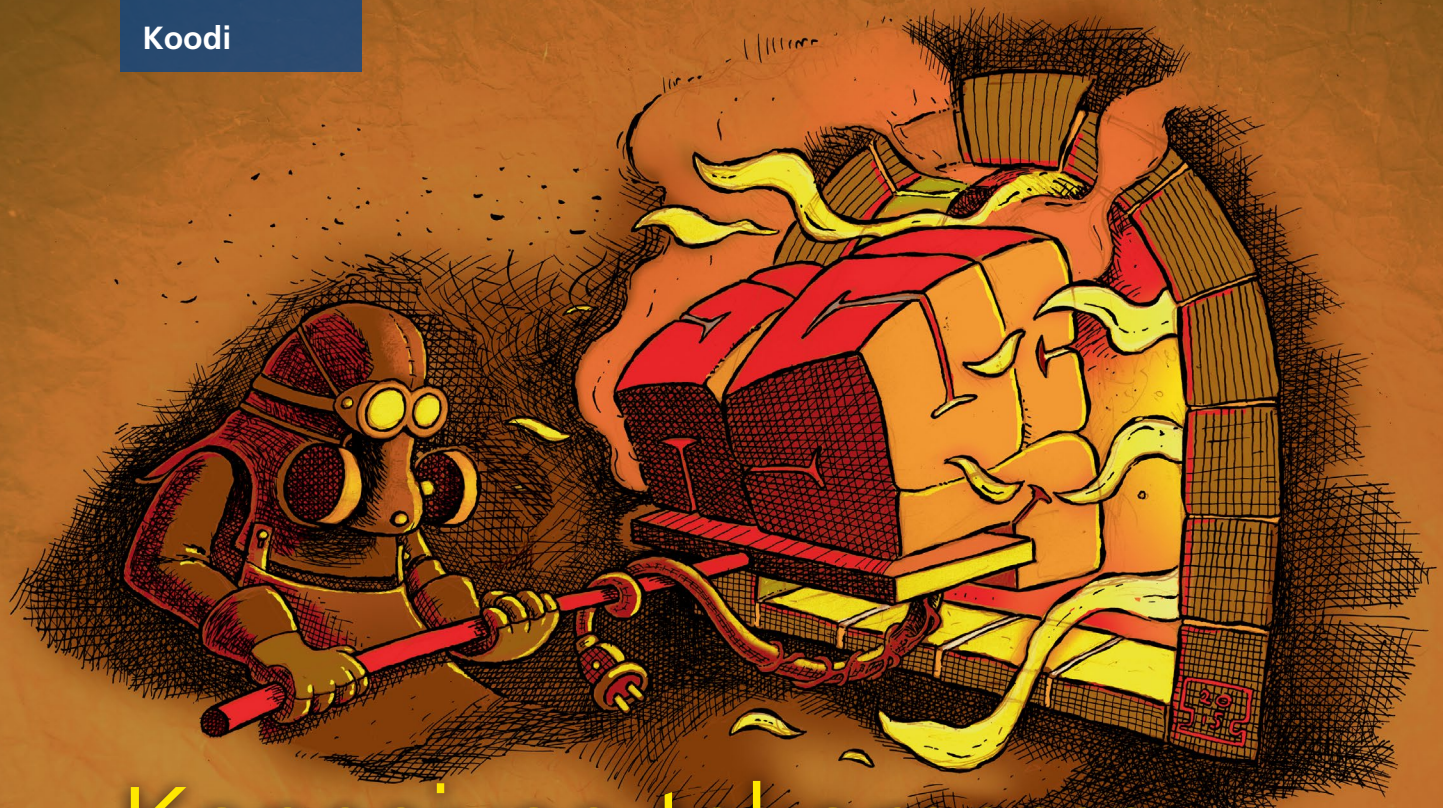
Kiinnostavaa Pop up -olohuoneessa oli tapahtumien vaihteleva sisältö, koska jokainen päivä oli täysin erilainen. Lisäksi oli mielenkiintoista nähdä, löytäisivätkö paikalliset opiskelijat Olohuoneen, sekä tulisivatko samat ihmiset uudestaan eri tapahtumiin ja illanviettoihin.

Koko Otaniemi ei kuitenkaan löytänyt paikalle vajaan kolmessa viikossa. Tapahtumiin tuli yleensä järjestäjätahojen kavereita, heidän kohdeyleisöään sekä satunnaisia ohikulkijoita. Innokkaimmat olohuonekävijät olivat kuitenkin



Vj-jamien aikana kuka tahansa sai kokeilla videoklippien pyörittämistä Resolume 4 -ohjelmalla.
Kuva: Essi Laurila

Skrolli-illan artikkelin ja taltioidut videot löydät Skrolli.fi-sivustolta Artikkelit-kansiosta tai tämän linkin naputtelemalla: www.skrolli.fi/2015/05/skrolli-live-suunpieksantaa-pelialasta-ja-tekniikasta/.



Koneaivoa takomassa

Neuroverkkojen mysteerit auki

Neuroverkot tekevät hämmästyttäviä asioita. Ne tunnistavat ihmisiä kasvonpiirteistä, ohjaavat itsestään ajavia autoja ja tuottavat taidetta. Ne eivät kuitenkaan ole täysin käsittämättömiä taikalaatikoita – kuolevaisellakin on aivan hyvät edellytykset ymmärtää niiden periaatteita.

Teksti: Ville-Matias Heikkilä Kuvat: Tapio Lehtimäki, Mitol Meerna, Ville-Matias Heikkilä

Useimmat tavanomaiset tietojenkäsittelytehtävät eivät kaipaa neuroverkkoja avukseen. Viivanpiirtoa, varmuuskopioiden automatisointia tai toimintapelin mekaniikkaa toteuttava ohjelmoija pystyy analysoimaan ongelman täydellisesti omassa mielessään. Hän voi helposti käydä kaikki mahdolliset poikkeustilanteet systemaattisesti läpi ja järkeillä niille pätevät tai jopa optimaaliset ratkaisut. Kaikki tehtävät eivät kuitenkaan ratkea näin helposti – monien ongelmien kohdalla analyttimen järki saattaa pettää täysin.

Otetaanpa esimerkiksi pelit. Pelimekaniikka voi olla hyvin helppo toteuttaa, mutta entäpä peliä hyvin pelaava tekoäly? Tekoälyn ohjelmoija päätyy helposti kokeilemaan melko umpimähkäisesti erilaisia parametreja, kaavoja, apumuuttujia ja ehtolauseita. Tällaiseen tilanteeseen jouduttuaan ohjelmoijan onkin usein aiheellista kysyä itseltään, olisiko järkevää kokeilla neuroverkkoa tai jotain muuta automaattisesti säätyvää mekanismia.

Neuroverkon toimintaperiaate

Yksinkertaisia toimintapelejä pelaavat neuroverkot ovat melko helppoja ymmärtää ja toteuttaa, ja ennen kaikkea niiden lopputulokset ovat havainnollisia. Niitä onkin tästä syystä näkynyt monissa viraalivideoissa, jotka tutustuttavat ihmisiä neuroverkkojen periaatteisiin.

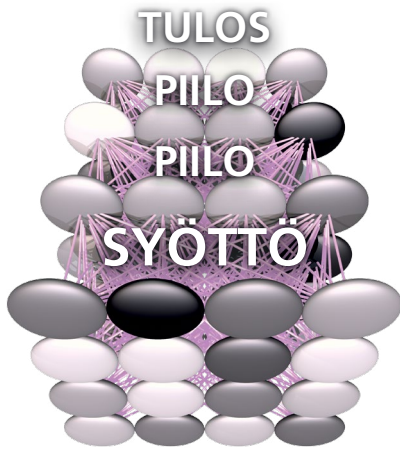
Keinotekoinen neuroverkko on hyvin karkeasti määriteltynä matemaattinen malli tai tietokoneohjelma, joka jäljittelee luonnollisia neuroverkkoja, elävien olentojen hermojärjestelmiä. Neuroverkon perusyksiköstä käytetäänkin biologiasta tuttua nimitystä *neuroni*. Yksittäisen neuronin tila kuvataan tietokoneessa yleensä yhdellä liukuluvulla – se on siis huomattavasti yksinkertaisempi kuin elävä hermosolu.

Neuronit on useimmissa verkoissa aseteltu kerroksiksi (*layers*), joista oleellisimpia ovat syöttö- ja tulostuskerrokset. Nokian matopelin kaltaista peliä pelaavalla verkolla voisi olla esimerkiksi 7×7 neuronin syöttökerros, joka esittää madon pään ympäristön siten, että tyhjiä ruutuja vastaa luku 0, esteitä luku 1 ja syötäviä hedelmiä -1.

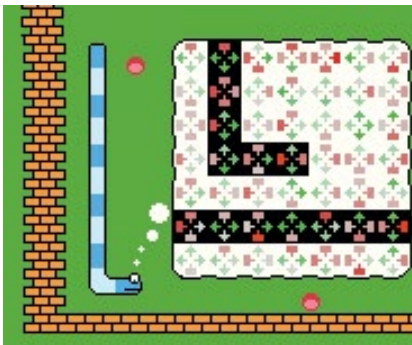
Tulostuskerros puolestaan voisi koostua neljästä neuronista, jotka vastaavat madon neljää mahdollista liikkumissuuntaa. Tulostuskerrosta tulkittaisiin siten, että verkko haluaa madon etenevän siihen suuntaan, jota vastaava neuroni ”loistaa kirkkaimmin”.

Neuronien välillä on linkkejä, joiden biologinen vastine on *synapsi*. Suoraviivaisinta linkitys on ns. eteenpäin syöttävissä verkoissa (*feed-forward network*), joissa yhden kerroksen neuronit linkittyvät aina seuraavan kerroksen neuroneihin ja data kulkee aina yhteen suuntaan.

Kullakin synapsilla on yleensä liukulukuna kuvattu painokerroin, joka ilmaisee sen, kuinka paljon sen lähtöneuronin tila vaikuttaa kohdeneuronin tilaan. Useimmiten neuronin tila-arvoksi asetetaan painotettu summa siihen linkittävien neuronien arvoista. Monesti tulosta vielä ”vahvistetaan” aktivaatiofunktiolla, joka voi olla esimerkiksi maksimi tai tangentti. Aktivaation tehtävä on samantapainen kuin transistoreilla mikropiireissä: se auttaa verkkoa valitsemaan useiden vaihtoehtojen välillä. Painotettua summaa ja



Eteenpäin syöttävän neuroverkon rakenne.



Piilokerrokseton matopeliverkko kytkee pelimaailman ruudut suoraan liikkumissuuntiin.

aktivaatiofunktioita käyttävä neuronirakenne on nimeltään *perseptroni*.

Syöttö- ja tulostuskerroksen välisiä kerroksia kutsutaan piilokerroksiksi (*hidden layers*). Niitä ei aivan kaikissa tehtävissä tarvitse, mutta tällöin on kyse pikemminkin lineaarifunktiosta kuin varsinaisesta neuroverkosta.

Matopeliä jotenkuten pelaavan verkon saa toimimaan ilman piilokerroksiakin. Tällöin verkon voi kuvata yksinkertaisesti synapsitaulukkona, jossa on kullekin pelinäkömään pikselille neljä kerrointa. Nämä kertoimet osoittavat, kuinka paljon madon etenemistodennäköisyys kuhunkin suuntaan muuttuu siitä, jos kyseisessä ruudussa on jotain muuta kuin tyhjää. Aiemmin mainituilla syöttö- ja tulostuskerroksen neuronimäärillä synapsimääräksi tulee kaikkiaan $7 \times 7 \times 4$ eli 196, mikäli kerrokset kytketään toisiinsa kokonaan.

Se oppii itse!

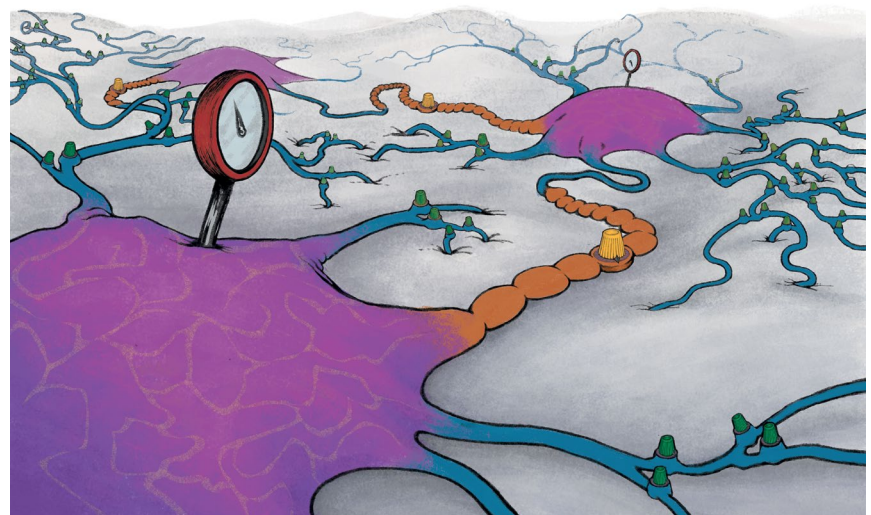
Vaikka neuroverkkojen painokertoimia voi säätää käsinkin, on niiden varsinainen juju säädön automatisoinnissa eli verkon koulutuksessa.

Koulutustavat voidaan jakaa ohjattuihin (*supervised*) ja ohjaamattomiin

(*unsupervised*). Ohjattua tapaa voidaan käyttää, jos tarjolla on runsaasti ”esipureksittua” dataa, jossa esimerkiksi otteet on yhdistetty haluttuihin tulosteisiin. Peliä pelaavassa verkossa tällaisena datana voitaisiin käyttää esimerkiksi hyvien ihmispelaajien reaktioita erilaisiin pelitilanteisiin. Ohjaamattomassa verkossa valmiita ei esimerkkejä ole.

Pelejä pelaaviin verkkoihin sopiva ohjaamaton koulutustapa on *neuroevoluutio*. Siinä annetaan satunnaisen verkkojen kilpailu keskenään, ja kunkin kierroksen jälkeen huonoiten pärjänneet verkot korvataan uusilla verkoilla, jotka on risteytetty parhaiten pärjänneistä. Useiden tuhansien risteytyskukupolvien kuluessa verkkojen pitäisi vähitellen säätyä aina vain paremmin tehtävän hallitseviksi. Neuroevoluutio on helppo toteuttaa ja siksi sopiva vaikkapa harjoitusprojektiksi. Esimerkkikuvan matopeliverkko on treenattu evoluutiolla.

Eteenpäin syöttävät verkot koulutetaan useimmiten takaisinvälityksellä (*back-propagation*). Tässä tekniikassa verkolle annetaan esimerkkisyöte ja verrataan sen antamaa tulostetta haluttuun tulosteeseen. Tämän jälkeen lasketaan kerros kerrallaan tulostuskerrokselta taaksepäin edeten, kuinka paljon kunkin synapsin painokerrointa on rukattava ja mihin suuntaan, jotta tulos olisi hieman lähempänä haluttua. Matemaattisemmin ilmaistuna tämä on gradientin seuraamista, ja siihen yleisesti käytetty algoritmi on nimeltään SGD eli *stochastic gradient descent*.



Millä tavoin synapsien säätönuppeja on pyöriteltävä, jotta mittarit osoittaisivat aina oikein? Ihmiseltä saattaisi mennä hermot, joten tehtävä on paras jättää koneelle.

Gradienttia orjallisesti seuraamalla juututaan helposti ns. paikallisiin maksimeihin: ei päästä mäennyppylältä sen vieressä olevalle vuorelle, koska välissä on alamäkeä. Ongelmaa voi ratkaista satunnaistekijöillä. SGD:ssä satunnaisuutta tuo koulutusdatan läpikäyntijärjestyksen arpominen, ja lisäksi verkon neuroneista voidaan kytkeä tietty satunnainen osuus pois päältä kullakin kierroksella.

Verkko voi luoda kategoriansa myös itse, jolloin se soveltuu esimerkiksi tavallisen ja epätavallisen erottamiseen toisistaan. Teuvo Kohosen itseorganisoituvat kartat toimivat näin. Kaikissa neuroverkoissa ei myöskään ole erillistä koulutus- ja käyttömodia, vaan ne oppivat käytön aikana samaan tapaan kuin biologiset hermostot. Tämäntyyppiset verkot koostuvat muistavista neuronyksiköistä, joiden tehtävänä on oppia ennustamaan niihin tulevia signaaleja. Jos ennuste ei vastaa todellisuutta, yksikkö voi vaikkapa ”ylälättä” ja antaa siitä signaalin eteenpäin.

Kuvantunnistusverkot

Hahmontunnistus on ongelmatyyppi, jossa neuroverkot pääsevät loistamaan. Kuvantunnistusverkoilla on myös äskettäin havaittu häkellyttäviä graafisia kykyjä, kun niiden on annettu tuottaa takaisinvälityksen kautta omia ”hallusinaatioitaan”.

Eräänlainen kuvantunnistusohjelmistojen vuotuinen kuninkuusmitelmä on ILSVRC (*Imagenet Large Scale Visual Recognition Challenge*). Kukin kilpailuun osallistuva ryhmä toimittaa siihen ohjelman – tyyppillisesti esikou-



VGG-tiimin 16-kerroksinen verkko vuodelta 2014.

lutetun neuroverkon – jonka tehtävänä on määrittää sille annetuista kuvista, mitä niissä on. Luokittelukategorioita on kaikkiaan tuhat erilaista, ja niihin kuuluu mm. satoja eläinlajeja ja arkielämän esineitä.

Viime vuosina hyvin menestyneet verkot ovat usein olleet ns. syviä konvolutiivisia verkkoja. Syvyys tarkoittaa sitä, että verkossa on paljon piilokerroksia – jopa useita kymmeniä. Tämä antaa verkolle tilaa muodostaa varsin pitkälle vietyjä abstraktioita näkemästään. Konvolutiivisuus puolestaan tarkoittaa kytkentätapaa, jossa kerroksen kuhunkin neuroniiin vaikuttaa edellisessä kerroksessa samassa kohti oleva 3x3, 5x5 tai useamman neuronin tai neuronipinon rypäs. Tällä tavoin kytketyt kerrokset toimivat pitkälti samaan tapaan kuin kuvankäsittelyohjelmien konvolutiiviset suotimet.

Monet ILSVRC:ssä kilpailleet ver-

kot ovat vapaasti ladattavissa ja käytettävissä ei-kaupallisiin tarkoituksiin. Googlen ja Oxfordin yliopiston menestyksekkäät verkot ovat ajettavissa Caffé-ohjelmistolla, joka on etenkin syvien verkkojen tutkimukseen kehitetty avoimen lähdekoodin ohjelma. Caffé panostaa nopeuteen ja osaa hyödyntää CUDA-rajapinnan kautta mm. grafiikkasuorittimien rinnakkaislaskentaa. Ohjelmointirajapintoja on tarjolla C++:lle ja Pythonille, ja etenkin Python-esimerkkikoodia on varsin helppo löytää.

Oheessa on kuvattu Oxfordin yliopiston Visual Geometry Groupin, vuonna 2014 käyttämä 16-kerroksinen verkko, joka yksinkertaisesta rakenteestaan huolimatta pärjäsi kisassa erinomaisesti. Useimmat sen kerrokset ovat 3x3-konvoluutiokerroksia, jotka leikkaavat samalla summista negatiiviset arvot nolliksi. Aina muutaman konvoluutiokerroksen jälkeen on *maxpool*-kerros, joka ottaa kustakin 2x2:n tai 3x3:n neuronin rypästä talteen suurimman arvon. Maxpool-kerrokset vähentävät verkon herkkyyttä kuvien merkityksettömille asemointieroille. Viimeistä maxpool-kerrosta seuraa kolme kerrosta, joiden jokainen neuronin on kytketty jokaiseen edellisen kerroksen neuroniiin. Näistä viimeisessä on tuhat neuronaa – yksi kutakin luokitteluryhmää varten.

Verkon vaatima tallennustila riippuu varsin suoraan sen synapsimäärästä. Googlen kuvantunnistusverkko Googlenetin mallidata vie levyltä reilut 50 megatavua, kun taas VGG:n verkkojen koot ovat olleet useita satoja megoja. Muistiin ladattu verkko vaatii lisäksi tilaa neuronikohtaiselle datalle.

Mysteereihin valoa uneksimalla

Neuroverkkojen rakenteita ylläpitävä algoritmiikka on varsin yksinkertaista, ja sen voi halutessaan ohjelmoida melko kivuttomasti itsekin. Se kuvaa kuitenkin pelkän raaka-aineen, josta varsinainen logiikka muotoutuu koulutuksessa. Tämä logiikka on huomattavasti vaikeaselkoisempaa, optimointialgoritmit kun eivät kunnioita ihmisinsinöörien modularistisia ihanteita. Sitä on ollut vaikea kuvata asiantuntijoidenkaan ymmärtämässä muodossa.

Jos kuvantunnistusverkon tietyn ker-

roksen neuronitilat visualisoidaan suoraan 2D-kuvaksi, tuloksena on yleensä epämääräisiä läikkiä. Nämä läikät ovat kuin aivojen magneettikuvia – tutkija saattaa pystyä esittämään niiden perusteella valistuneen arvauksen, mitä verkossa tapahtuu, mutta maallikolle ne eivät sano mitään. Huomattavasti havainnollisempi visualisointitapa on Googlen tutkijoiden melko äskettäin keksimä, takaisinvälitysmekanismia hyödyntävä menetelmä.

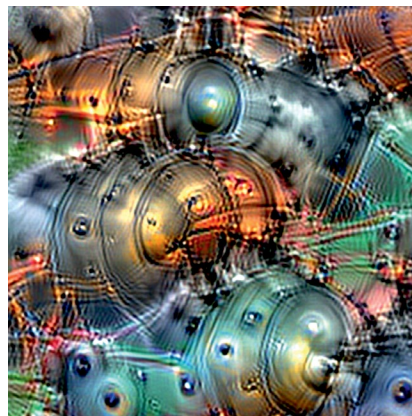
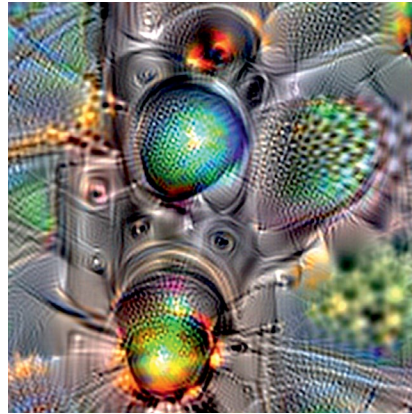
Miltä tietokone näyttää neuroverkon mielestä? Syötetään verkolle aluksi kohinaa, josta se saattaa tunnistaa mitä tahansa. Tämän jälkeen lasketaan, kuinka kuvaa pitää muuttaa, jotta sen tunnistus olisi hieman lähempänä tietokonetta. Pöytätietokoneen tunnistava tuloskerroksen neuronin ILSVRC12-kategorioiden mukaan numero 527, joten sen ulostulo on saattava kasvamaan.

Koulutusikäytössä takaisinvälitykseen kuuluu kaksi vaihetta: ensin lasketaan, kuinka kunkin edelliskerroksen neuronin ulostulo muuttuu, jos tuloskerroksen neuroneille tehdään halutut muutokset, ja sen jälkeen ruokataan synapsikertoimia näiden muutosten mukaan. Tässä tapauksessa jälkimmäinen vaihe voidaan unohtaa, sillä meitä kiinnostaa vain, kuinka syöttökerroksen neuronien tilat muuttuvat tuloskerroksen muutoksen myötä. Kun muutoksia kasataan silmukassa alkuperäisen kuvan päälle, kohinan seasta paljastuu vähitellen verkon käsitys tietokoneesta.

Entäpä, jos halutaan tietää, mitä kuvantunnistusverkon mikäkin kerros tekee? Kun tietyn kerroksen neuronien arvoja kasvatetaan sitä enemmän, mitä suurempia ne ovat ennestään, saadaan takaisinvälityksen avulla vahvistettua kuvasta niitä yksityiskohtia, joihin tämä nimenomainen kerros kiinnittää huomiota. Alimmilta kerroksilta paljastuu tyypillisesti perustason kuvankäsittelyoperaatioita kuten väriarvojen etsintää, kun taas ylimmät kerrokset tuottavat jo selvästi tunnistettavia eläimiä, esineitä ja rakennuksia. Tämä kuvan piirteiden vahvistamistapa sai Googlen tutkijoilta nimen *deep dream*.

Taiteelliset ulottuvuudet

Googlen kuvantunnistusverkon tuottamat *deep dream* -kuvat saivat kesällä 2015 aikaan netti-ilmion, jossa samaa



VGG:n verkolta maaniteltiin esiin kuvia, jotka sen mielestä tunnustuvat erityisen vahvasti tiettyihin aiheisiin: hedelmäpeli, liikennevalo, kynttilä, höyryveturi.



Tällaisia pintoja syntyy, kun VGG:n verkolle annetaan oppaaksi kuvat lahottajasienistä ja avaruusaluksista.

Googlen esimerkkikoodia sovellettiin sellaisenaan kaikenlaisiin kuviin. Koska ILSVRC:n tuhanteen tunnustus-kategoriaan kuuluu useita eri koirarotuja, syntyy Imagenet-datalla treenatuille verkoille herkästi taipumus etsiä kuvista yksityiskohtia, jotka erottavat näitä rotuja toisistaan. Niinpä deep dream -kuviinkin generoitui koiranpäitä suorastaan kyllästymiseen asti.

Kuvantunnistusverkoissa on kuitenkin huomattavasti enemmän taiteellista potentiaalia, kuin mihin kesän järkytysmeemi ehti ylittää. Yksityiskohtien vahvistumista voi ohjata lukematto-

milla eri tavoilla, verkkoja voi treenata omilla materiaaleilla ja niitä voi yhdistää moniin muihin tekniikoihin. Mahdollisuuksissa on hädin tuskin päästy edes raapaisemaan pintaa.

Hyvä tapa saada aavistus verkon luovuudesta on antaa sen "uneksia" eli sukeltaa omien tuotostensa sisään: kun kuvan yksityiskohtia on vahvistettu deep dream -algoritilla, sitä skaalataan hieman ja skaalattu kuva syötetään takaisin algoritmillemme. Kun tätä jatketaan silmukassa, syntyy fraktaalizoomeria muistuttava animaatio, jossa kuvan yksityiskohdista paljastuu

loputtomiin uusia yksityiskohtia.

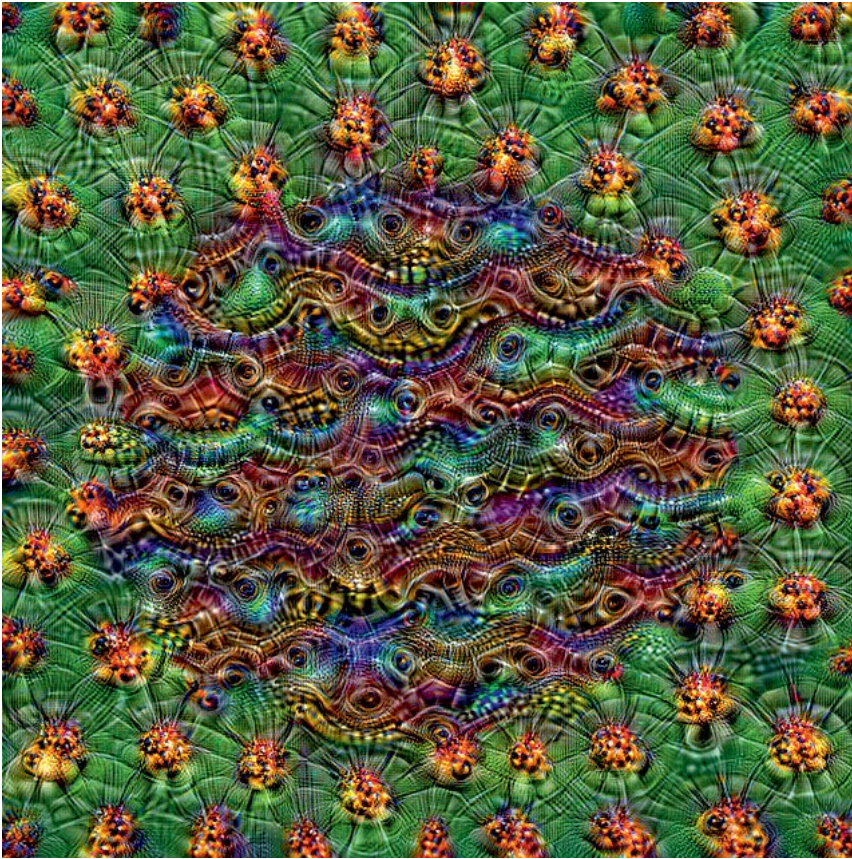
Vapaa uneksinta paljastaa kuitenkin vain pienen osan verkon mielikuvitusmaailmasta, sillä tietyt elementit korostuvat siinä suhteettoman paljon. Kuten ihmismieli on kärkeä tunnistamaan kasvoja jopa epämääräisistä läikistä, myös neuroverkot kehittävät herkästi vastaavanlaisen taipumuksen. Unien sisältöön voi vaikuttaa rukkaamalla kaavoja, jotka määräävät, minkä neuronien arvoja voimistetaan ja kuinka painokkaasti.

Helppo tapa ohjata uneksintaa on antaa verkolle opaskuva (*guide*), jossa on halutunlaisia piirteitä. Opaskuvan tuottamia neuroniarvoja käytetään uneksinnan aikana painokertoimina neuroniarvojen muutoksille. Näennäisestä hallittavuudestaan huolimatta tulokset saattavat kuitenkin olla hyvin odottamattomia: esimerkiksi eräs avaruusaluksista esittävä opaskuva tuottaa gigermäistä lonkeroseinämää. Avaruusalus ei kuulu Imagenet-kategorioidiin, joten verkko ilmeisesti yhdisti kuvan mereneläviin.

Yleiset kuvantunnistusverkot ovat uneksintakokeilujen perusteella erityisen kyvykkäitä pienissä orgaanisissa yksityiskohdissa, jotka ovat toisinaan jopa häiritsevän realistisia. Pinnat, varjostukset ja palasten yhteenliittämiset ovat verkkojen vahvuusalueita, kun taas perspektiivit ja rakennetut muodot ovat ainakin Imagenet-kuvista oppinsa saaneille verkoille paljon vaikeampia. Kokeilemistani esitreenuista verkoista ainoastaan Googlenet tuottaa uneksiessaan maisemakuvia, ja niissäkin saattaa kuvan alareunaan ilmestyä toinen taivas.

Deep dream -tyyppiset algoritmit saattavat hyvinkin ruveta valtaamaan alaa taiteilijoiden apureina, jotka väkertävät yksityiskohtia ja varjostuksia. Taiteilija saattaa piirtää esimerkiksi luonnoksen, jossa on eri alueet merkitty sen mukaan, mitä niille halutaan tehtävän. Reaaliaikaisemmassa vaihtoehdossa nämä eri käsittelytavat voisivat olla erilaisia "siveltimiä". Lisäksi esimerkiksi äskettäisessä Siggraph-konferenssissa oli artikkeli kuvan valaistuksen muuttamisesta neuroverkoilla. Mahdollisuuksia on siis moneen suuntaan.

Nykyisten kuvantunnistusverkkojen yhdellä kertaa käsittämä näkökenttä on yleensä vajaat 256×256 pikseliä.



Kuvan keskiosaan on uneksittu sisältöä eri opaskuvalla kuin reunoille. Rajakohdassa näkyy, kuinka verkko pyrkii sovittamaan erityyppisiä rakenteita yhteen.

Kuvan voi toki rakentaa useissa pala-sissa ja skaalautasoissa, mutta tällöin eri vaiheet saattavat ikävästi sotkea toisiaan. Myös osa tämän artikkelin kuvista on tästä syystä varsin karkeita.

Omaa verkkoa opettamaan

Esitreenuilla verkoilla on mukava leikkiä, mutta silloin ollaan sidoksissa niihin kuviin ja kategorioihin, joilla verkko on alun perin koulutettu. Tästä syystä olisi mukava kouluttaa aivan oma verkko tai vaikka useampikin, joiden taiteelliset taipumukset ovat yhtä persoonalliset kuin kouluttajansakin.

Koska monimutkaisten verkkojen treenaukseen saa helposti uppoamaan paljon laskentaresursseja, lienee aiheellista vältellä sooloilua ja tukeutua valmiisiin malleihin. Tutkijat kuitenkin ovat vuosikausien ajan viilailleet kuvantunnistusverkkojaan.

Mutta mitkä mallit ovat hyviä taidekäyttöön? Hyvä menestys ILSVRC:n tunnistustehtävissä ei välttämättä tarkoita, että sen tuottama takaisinvalityskuvasto olisi mitenkään esteettistä. Latailin vertailun vuoksi Caffé-wikin *model zoosta* joukon erilaisia verkkoja, jotka oli valmiiksi koulutettu samalla

Imagenet-datalla, ja annoin jokaisen uneksia jonkin aikaa samalla opaskuvien sarjalla. Tein ajoista myös Youtube-videon (*Comparing the hallucinations of four similarly trained neural nets*).

Vaikka Googlenet ja VGG:n CNN-S-verkko tuottivatkin esteettisesti mielenkiintoisinta jälkeä, päätin valita ensimmäiseksi koulutettavakseni Singaporen yliopistossa kehitetyn NIN-mallin (*Network in Network*). Pääasiallisena syynä valintaani oli resurssinkäyttö: verkkoa on nopeampi ajaa kuin kilpailijoitaan, ja sen myös luvataan oppivan nopeasti. Käytössäni ei ole CUDA:n tukemaa laskentarautaa, enkä halua odotella ensikokeilujeni tuloksia kuukausitolkulla.

NIN poikkeaa tämänhetkisestä kuvantunnistusverkkojen valtavirrasta siinä, että se käyttää lineaaristen konvoluutiosuotimien sijaan pieniin neuroverkkoihin pohjautuvia suotimia. Singaporelaiset uskovat, että pienoisverkkojen ilmaisuvoima lineaarisiin funktioihin nähden antaa NIN-verkolle kokoonsa nähden vahvan abstraktiokyvyn, ja se päihittääkin tunnistustehtävissä monet isommat verkot.

Kuvantunnistusverkon ohjattu kou-

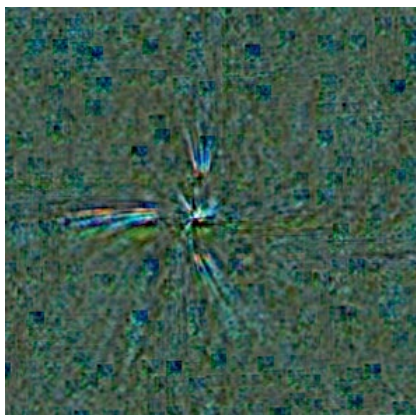


Network in Network -verkon rakenne.

lutus tarvitsee valtavan määrän luokiteltua kuvamateriaalia. Imagenet tarjoaa yksinkertaisen tekstitiedoston, jossa on hulppeat 14 miljoonaa kuvauria yhdistettynä kymmeniintuhansiin Wordnet-indekseihin, jotka toimivat luokittelujen pohjina. Imagenetin luokittelussa on silti ammottavia aukkoja – esimerkiksi löytänyt siitä havupuita lainkaan. Tein siksi sen avuksi skriptin, joka täydentää aukkoja Flickr-hakujen avulla.

Ajattelin aloittaa kevyesti opettamalla verkon tunnistamaan muutamaa erisientä ja marjaa – kaikkiaan noin 12 000 kuvan joukko. Valikoimaa voisi laajentaa myöhemmin muunlaisiin kohteisiin, jos tulokset näyttävät lupaavilta.

Kun treenauskuvat on haettu tiedostoihin, niistä muodostetaan LMDB-tietokanta Caffén mukana tulevalla *convert_imageset*-työkalulla. Avuksi tarvitaan myös tekstitiedosto, jossa on kullakin rivillä tiedoston nimi ja sen

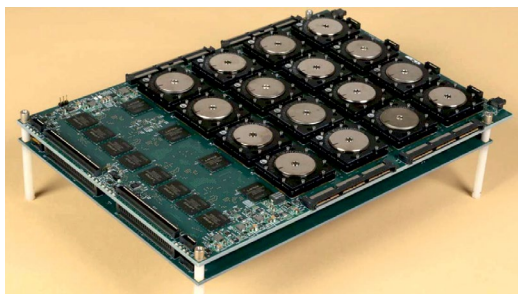


NIN-verkon unet eivät ole vielä sadantuhannen koulutusiteraationkaan jälkeen kovin tolkullisia.

sisällön luokitteleva kategorianumero välilyönnein erotettuna. Samalla työkalulla kannattaa muodostaa myös testauksessa käytetty suppeampi *val*-kuvasto, jolla verkon tunnistuskyvyn kypsyminen voi seurata. vMolemmista kuvastoista lasketaan *compute_image_mean*-työkalulla keskiarvo, joka vähennetään kaikista verkolle syötettävistä kuvista. Jos keskiarvokuvaa ei lasketa, verkon parametrit voivat ruveta koulutusvaiheessa puoltamaan äärettömyyksiä päin, jolloin koko ajo epäonnistuu.

Caffe-oppaat neuvovat yleensä kouluttamaan verkot puhtaalta pöydältä. Tässä on kuitenkin omat ongelmansa: suuri osa laskenta-ajasta menee siihen, kun verkko opettelee näkemisen alkeet, ja jos koulutus menee pieleen epäonnisten valintojen vuoksi, verkko ei koskaan opi edes niitä. Siksi voi olla mielekäästä käyttää pohjana esitreennattua verkkoa. Jos verkossa ennestään oleva kuvasto häiritsee, kouluttaja voi alustaa verkon kerroksia yksi kerrallaan ylhäältä alaspäin, kunnes se ei enää uneksi mitään tunnistettavaa. Valmiin verkon täydennys- ja uudelleen koulutuksessa on kuitenkin omat hankaluutensa siinäkin.

Neuroverkon luominen on hyvin erilaista kuin perinteinen ohjelmistonkehitys. Sitä voisi verrata vaikkapa siihen, mitä kasvien kasvattaminen on verrattuna koneiden rakentamiseen. Kouluttaja tarjoaa verkolle sopivaa kasvualustaa ja ravintoa ja ehkä välillä hieman karsii tai kitkee. Toki neuroverkkojakin voi rakentaa kouluttamalla sen osia erikseen ja liittämällä niitä yhteen, mutta etenkin ns. syväoppimiseen (deep learning) kuuluu se, että



DARPA:n Synapse 16 -neurolaskentalauta mallintaa rinnakkain 16 miljoonaa neuronin ja 4 miljardia synapsia. Kuva: DARPA

myös ison mittakaavan rakenteet muotoutuvat verkkoon ilman insinööriä.

Lähemmäksi ihmisiä

Tavanomaiset eteenpäin syöttävät verkot ovat sielukkaista taipumuksistaan huolimatta aika paljon lähempänä tietokoneohjelmia kuin ihmisiä. Kun verkko on kertaalleen koulutettu, se antaa samalla syötteellä joka kerta saman tuloksen – ellei siihen ole vartasta vasten rakennettu satunnaisuutta. Verkon kehitystyö ja käyttö ovat selvästi erillisiä asioita, eikä koulutukseen tarvitse välttämättä käyttövaiheessa palata enää koskaan. Syväkin eteenpäin syöttävä verkko on oikeastaan ”tyhmempi” kuin tietokone: sen voi rakentaa vaikkapa analogiseksi piiriksi, johon ei kuulu yhtäkään Turingtäydellistä osaa.

Biologiset hermostot eroavat yksinkertaisimmista neuroverkkomalleista etenkin siinä, että ne ovat luonteeltaan ajallisia. Ne reagoivat muuttuviin tilanteisiin ja sopeutuvat ajallisiin signaalinmuutoksiin. Keinotekoiseen neuroverkkoon saa aikaulottuvuuden esimerkiksi linkittämällä joitakin sen neuroneita edellisen ajokierroksen neuronitiloihin, jolloin syntyy *rekurrentti* verkko (RNN). Toinen vaihtoehto on antaa verkolle muistisoluja, joiden sisältöä se pystyy lukemaan ja kirjoittamaan.

Ajalliset verkot soveltuvat toisaalta esimerkiksi puheen, musiikin tai tekstin tunnistamiseen ja tuottamiseen, toisaalta taas ohjaustehtäviin, joita on esimerkiksi robotiikassa ja teollisuudessa. Ajalliset verkot vaativat kuitenkin usein toisenlaisia koulutusmenetelmiä, koska suhde syötteeseen ja tulosteen välillä ei ole enää niin suoraviivainen.

Kun verkolla on oma muisti, joka pystyy korvaamaan synapsipainotusten tehtäviä, se ei välttämättä tarvitse

erillistä koulutusvaihetta vaan oppii toimiessaan. Tällaisia verkkotyyppisiä ovat esimerkiksi LSTM (*Long Short-Term Memory*) ja PBWM (*Prefrontal Cortex Basal Ganglia Working Memory*), joista jälkimmäisen esikuvana on varsin suoraan ihmisajon otsalohko. Nämä rakenteet pystyvät jäljittelemään yhtä lailla perinteisen tietokoneen

kuin biologisten aivojenkin toimintaa.

Erilaisia neuroverkkotyyppisiä on lukemattomia, ja rajatapauksissa voi olla epäselvää, milloin kyseessä on neuroverkko ja milloin ei. Esimerkiksi Googlen kääntäjä on tilastollinen itseoppiva järjestelmä samaan tapaan kuin monet neuroverkot, mutta sen tallennusmalli ei perustu synapsityyppeihin säätyviin parametreihin, joten sitä ei pidetä neuroverkkona.

Erityyppisten verkkojen toteuttamiseen on tarjolla erilaisia ohjelmistopaketteja. Esimerkiksi Caffe on tarkoitettu erityisesti syväoppiviin eteenpäin syöttäviin verkkoihin, vaikka sen pystyykin jotenkuten taivuttamaan myös rekurrentteihin ja muistaviin verkkoihin. Caffe kuitenkin laajentaa jatkuvasti alaansa, joten sen opettelu ei ole välttämättä mikään huono päätös aivosimulaatiostakaan kiinnostuneille.

Laitteistopuolella neurolaskentaan käytetään useimmiten joko tavallista suorittinta tai GPU:ta, mutta erityisesti sitä varten suunnitellun laskentalaitteiston voi olettaa yleistyvän tulevaisuudessa. Nanometriluokan ”neuro-morfisten” piirien kehittämiseksi on käynnissä useita tutkimusprojekteja mm. HP:llä ja IBM:llä.

Neurolaskenta ei kuitenkaan vaadi erikoista rautaa. Esimerkiksi moniin mobiililaitteisiinkin kuuluu neuroverkkopohjaisia hahmontunnistimia. Samalla Googlen kaltaiset tahot käyttävät ihmisiä yhä laajemmalla koneistolla, jossa neuroverkot ovat yhä tärkeämmässä osassa. Tämän vuoksi tavallisenkin kansalaisen on hyvä oppia ymmärtämään neuroverkkoja, vaikkapa sitten kuvantunnistusverkoilla leikkimällä. Ja kun leikkiessä saa samalla näkökulmaa myös omien aivojensa toimintaan, niin kovin hyviä syitä jättää tutustumatta neuroverkkoihin ei enää ole. 🤖



Vim

Vim on tehokas tekstieditori, jonka toiminta perustuu ikivanhoihin ideoihin ja hyväksi havaittuihin perinteisiin. Ehkä juuri siksi se on niin ylivoimainen verrattuna nuorempaan editorisukupolveen. Unohda siis lelueditorit ja ota käyttöön todellinen tehotyökalu.

Teksti: Teemu Likonen

Kuvat: Mitol Meerna, Teemu Likonen, Wikimedia Commons (D0ktorz, StuartBrady, Chris Jacobs), Jason Long

Jos tietokoneessasi on Unixin kaltainen käyttöjärjestelmä kuten GNU/Linux tai OS X, siihen on hyvin todennäköisesti asennettu tekstieditori nimeltä Vim. Se perustuu vuonna 1976 kehitettyyn Vi-editoriin, jota käytettiin alkuaikojen Unix-järjestelmissä.

Mutta miksi nykyaikaiset käyttöjärjestelmät raahaavat vieläkin mukanaan ikivanhoja Unixin ohjelmaraatoja?

Osasyynä on Unixin perinne ja standardi, mutta tärkeämpi peruste on se, että ohjelma on ihan oikeasti tarpeellinen ja tehokas nykyaikanakin.

Vi:lläkin kehitettiin aivan toisenlaiseen maailmaan. Alkuperäisen Vi:n kehittäjä Bill Joy käytti yksinkertaista päätelaitetta, jossa kirjaimet ropisivat ruudulle 300 baudin nopeudella. Nykyajan tekniikkaan verrattuna ruudun päivitys oli erittäin tahmeaa, ja se pakotti miettimään erikoisia ratkaisuja tekstin muokkausohjelmaan. Vi:n toiminta perustuikin siihen, että editorin piti olla tehokas yksinkertaisilla ja hitailla pääteyhteyksillä.

Alkuperäiset Vi:n kehittämisen lähtökohdat eivät tietenkään enää ole voimassa, mutta editori on hämmästyttävän tehokas tekstinmuokkaaja nykyaikaisissa. Rajallisia resursseja varten hiottu ohjelma on hyvä lähtökohta myös uudistuksille, koska perusasiat ovat varmasti kunnossa.

Vi:stä on aikojen saatossa tehty useita erilaisia versioita, ja niistä suurin ja kaunein on Vim, johon tämä artikkeli keskittyy. Vim sisältää alkuperäiseen Vi:hin verrattuna runsaasti uusia toimintoja ja on myös laajennettavissa. Tutkikaamme tarkemmin, millaisesta editorista on kyse.

Tilasta toiseen

Tavallisissa tekstieditoreissa näppäimistöä käytetään lähinnä vain tekstin tuottamiseen ja kursorin liikkuttamiseen. Erikoisemmat toiminnot haetaan hiiren avulla valikoista, tai joi-takin on sijoitettu myös toimintonäppäinten kuten *Ctrl*:n taakse.

Vim ei ole tavallinen editori. Sen toiminta perustuu eri tiloihin, joissa näppäimistö käyttäytyy eri tavoin. *Normaalitilassa* (normal mode) näp-



Ohjelmatyyppi	Tekstieditori
Ensimmäinen julkaisu	1.14 (2.11.1991)
Uusin versio	7.4 (10.8.2013)
Kehittäjä	Bram Moolenaar
Käyttöjärjestelmät	Linux, OS X, Windows ym.
Kotisivu	http://www.vim.org

päimistön kirjaimet, numerot ja muut merkit ovat komentoja, joilla esimerkiksi liikutaan tekstissä tai muokataan sitä. *Lisäystilassa* (insert mode) näppäimistöä käytetään tekstin kirjoittamiseen kuten editoreissa tavallisesti. Normaalitylissa näppäimellä *i* siirrytään lisäystilaan; takaisin normaalitylilaan palataan *Esc*-näppäimellä.

Erillinen tila tekstin muokkauskomennolle johtui alun perin siitä, että 70-luvun päätelaitteiden näppäimistöissä ei juuri ollut hyviä vaihtoehtoja tekstinmuokkauskomennolle. Samanlaisia rajoituksia ei nykykoneissa ole, mutta eri tiloihin perustuva tekstinmuokkaus toimii hienosti edelleen. Normaalitylissa näppäimistön merkkipalikoimaa hyödynnetään tehokkaasti, koska se on kokonaan varattu muokkauskomennolle ja tekstissä liikkumiseen.

Liikettä

Normaalitylan komennot voidaan jakaa kolmeen ryhmään: liikekomentoihin, operaattoreihin ja tekstivalintakomentoihin. Liikekomennolla siirretään kursoria tekstin seassa. Kursoria voi liikuttaa esimerkiksi yhden merkin, rivin, sanan, virkkeen, suljeparin, kappaleen tai kuvaruudun verran.

Tavallinen kursorin liikuttaminen sujuu tuttuun tapaan nuolinäppäimillä, mutta Vimissä on toinenkin, kenties tehokkaampi vaihtoehto, joka sekun on peräisin kaukaa menneisyydestä. Alkuperäisen Vi:n kehittäjän päätelaitteessa ei ollut erillisiä nuolinäppäimiä vaan kursoria liikutettiin näppäimillä *h*, *j*, *k* ja *l*. Ratkaisu voi tuntua kömpelöltä, mutta sen hyvä puoli on, että kättä ei tarvitse siirtää pois kirjainnäppäinten luota edes kursorin liikuttamiseksi. Ainakin Vimin tehokäyttäjille se kelpaa, sillä he pyrkivät kehittämään ergonomian ja muokkausnopeuden mahdollisimman pitkälle.

Liikekomentoihin voi kytkeä luku-

määrän, jolloin samaa komentoa toistetaan useita kertoja. Esimerkiksi *w* siirtyy yhden sanan verran eteenpäin mutta *3w* siirtyykin kolmen sanan verran. Taulukkoon 1 on kerätty tavallisia liikekomentoja.

Operaattoreita

Operaattoreiden avulla muodostetaan tekstinmuokkauskomentoja. Ne eivät toimi yksinään vaan tarvitsevat avukseen esimerkiksi liikekomennon, jolla ilmaistaan operaattorin vaikutusalue. Esimerkiksi *d* (delete) on poisto-operaattori, ja kun se yhdistetään liikekomentoon *\$*, saadaan komento *d\$*, joka poistaa tekstiä rivin loppuun saakka. Samalla tavalla käytetään muitakin operaattoreita, esimerkiksi kopiointi-operaattoria *y* (yank).

Poisto- ja kopiointikomennot siirtävät tekstin rekisteriin (vrt. leikepöytä), josta tekstin voi myöhemmin palauttaa. Rekisterissä oleva teksti liitetään komennolla *p* tai *P*. Taulukossa 2 on esimerkkejä poisto- ja kopiointikomennosta, ja taulukossa 3 on lueteltu muita operaattoreita.

Valintoja

Operaattoreita voi käyttää liiketoimintojen lisäksi myös tekstivalintakomentojen kanssa. Tekstivalintakomennot eivät yksinään tee mitään, mutta ne sisältävät ajatuksen tekstialueesta kuten virkkeestä, kappaleesta, lainauksesta tai muusta rakenteesta.

Esimerkiksi monissa ohjelmointikielissä ilmaistaan {}-merkeillä koodilohko, joka voi sisältää useita ohjelmointikielen lauseita. Vimissä tällaisen koodilohkon voi poistaa komennolla *da*, jossa *d* on poisto-operaattori ja *a* on koodilohkon valinta. Vastaavasti lainausmerkeillä rajattu tekstialue poistetaan komennolla *da*". Tekstikappale saadaan rivitettyä uudelleen, kun yhdistetään muotoiluoperaattori *gq* ja kappaleenvalinta *ap*.



h	vasemmalle
j	alas
k	ylös
l	oikealle
w	sana eteenpäin
b	sana taaksepäin
)	virke eteenpäin
(virke taaksepäin
}	kappale eteenpäin
{	kappale taaksepäin
o	rivin alkuun
\$	rivin loppuun
gg	puskurin alkuun
G	puskurin loppuun
5G	riville 5

Taulukko 1. Vimin normaalitylan liikekomentoja.

Poisto	Kopiointi	Vaikutusalue
dw	yw	seuraava sana
d2b	y2b	kaksi edellistä sanaa
d0	y0	rivin alkuun saakka
d\$	y\$	rivin loppuun saakka
d)	y)	virkkeen loppuun saakka
dG	yG	puskurin loppuun saakka
dd	yy	nykyinen rivi

Taulukko 2. Esimerkkejä poisto- ja kopiointikomennosta.

d	poista
y	kopioi
<	rivien siirto sarkaimen verran vasemmalle
>	rivien siirto sarkaimen verran oikealle
gu	pienet kirjaimet
gU	isot kirjaimet
gq	muotoilu (rivitys uudelleen)
!	muokkaus ulkoisen ohjelman avulla

Taulukko 3. Vim-editorin operaattoreita.

!	"	#	\$	%	&	'	()	0	*	=	{	}	Home
1	2	3	4	5	6	7	8	9	0	:	-	[]	^
Esc	Q	W	E	R	T	Y	U	I	O	P	Line Feed	Enter	Here is	
Ctrl	A	S	D	F	G	H	J	K	L	+	,		Rub	Break
Shift	Z	X	C	V	B	N	M	<	>	?	Shift	Repeat	Clear	
↑											↑			

Bill Joyn käyttämän ADM-3A-päätteen näppäimistö.

aw	iw	sana
as	is	virke
ap	ip	kappale
a"	i"	lainaus
a}	i}	{-merkeillä rajattu alue
a]	i]] -merkeillä rajattu alue
at	it	HTML-tagien pari

Taulukko 4. Tekstinvalintakomentoja: *a*-versio vaikuttaa koko alueeseen mutta *i*-versio vain sisältöön.

Taulukossa 4 esitellään muutama tekstinvalintakomento. Jokaisesta komennosta on olemassa kaksi eri versiota: taulukon ensimmäisessä sarakkeessa oleva *a*-alkuinen versio vaikuttaa koko tekstialueeseen mutta toisen sarakkeen *i*-versio vain alueen sisältöön. Esimerkiksi jos kursori on HTML-koodissa rakenteen `tekstiä` sisällä, komento *dat* poistaa sekä tagit että niiden sisällön, mutta komento *dit* poistaa pelkästään tagien välissä olevan alueen.

Tekstin valitsemiseen on olemassa myös visuaalinen tila (visual mode), joka on sama asia kuin muista editoreista tuttu tekstin maalaaminen hiiren avulla. Visuaalisia tiloja on kolme erilaista, ja ne käynnistetään komennoilla *v*, *V* tai *Ctrl+v*. Ensin mainittu valitsee tekstiä merkin tarkkuudella, toinen kokonaisia rivejä ja kolmas suorakulmaisen tekstilohkon. Kun tekstialue on valittu visuaalisen tilan avulla, siihen voi kohdistaa minkä tahansa operaattorin.

Komentorivi

Normaalitilassa kaksoispistekomento

(:) avaa ruudun alareunaan komentorivin, joka mahdollistaa pidempien komentojen kirjoittamisen. Myös tämän juuret ovat kaukana menneisyydessä, sillä osa komentorivin komennoista on peräisin ikivanhasta Unixin rivieditorista Ed.

Komentoriviä käytetään Vimissä paljon. Tekstin editoijan arkipäivä ovat tiedoston tallentaminen (*:w tiedosto.txt*), lataaminen (*:e tiedosto.txt*) sekä ohjelmasta poistuminen (*:q*). Usein tarvitaan myös *:s*-komentoa, jota toteuttaa etsi ja korvaa -toiminnon. Komento on monille tuttu Unixin sed-työkalusta.

```
:%s/etsi/korvaa/g
```

Todellinen voimatyökalu on *:g*, jolla voi suorittaa jonkin muun komennon kaikille riveille, jotka täsmäyvät tiettyyn säännölliseen lausekkeeseen (regular expression). Komento on kätevä esimerkiksi silloin, kun halutaan poistaa kaikki rivit, joilla esiintyy tietty teksti. Alla on siitä esimerkki; komentorivin lopussa oleva *d* on poistokomento.

```
:%g/teksti/d
```

Komentoriviä käytetään myös silloin, kun halutaan muuttaa editorin asetuksia. Erilaisia asetuksia on Vimissä runsaasti, ja niiden avulla editorin voi säätää toimimaan haluamallaan tavalla. Asetuksia muutetaan yleensä komennolla *:set*, jolle annetaan muutettavan asetuksen nimi ja mahdollisesti uusi arvo. Esimerkiksi komento *:set textwidth=72* asettaa rivin pituuden 72 merkkiin ja komento *:set backup* aset-

filetype plugin indent on
syntax enable

```
set nocompatible
set ruler
set wildmenu
set laststatus=2
```

Listaus 1. Omia suosikkiasetuksiaan voi kirjoittaa `~/.vimrc`-tiedostoon.

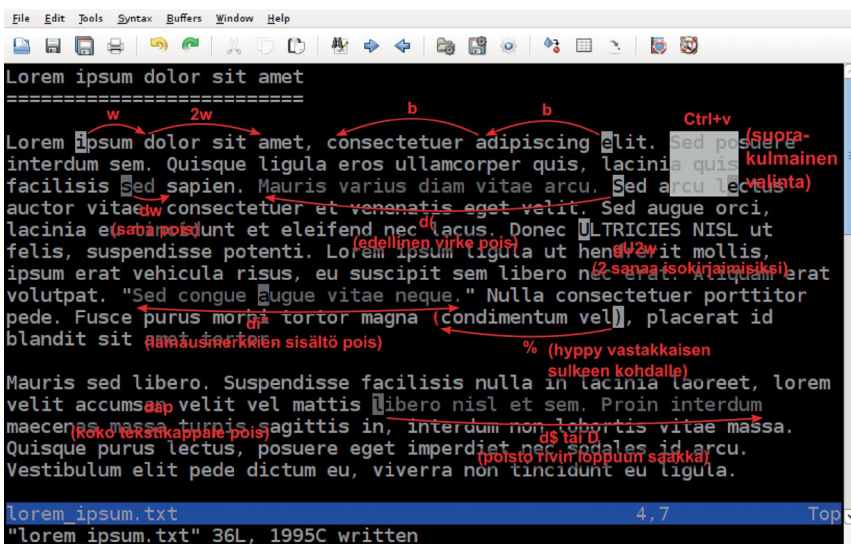
taa varmuuskopioinnin päälle, mikä seurauksena tekstitiedoston vanha versio säilytetään tallennuksen yhteydessä.

Tyypillisesti Vimin käyttäjä tallentaa suosikkiasetuksensa `~/.vimrc`-tiedostoon, joka ladataan automaattisesti ohjelman käynnistyessä. Hakemiston nimi `~/` viittaa käyttäjän kotihakemistoon Unixissa ja sen kaltaisissa käyttöjärjestelmissä. Asetustiedosto voi sisältää mitä tahansa Vimin komentorivin komentoja tai ohjelmointikielen lauseita. Listauksessa 1 on esimerkki `~/.vimrc`-tiedostosta, joka kytkee päälle eräitä hyödyllisiä ominaisuuksia.

Muita komentorivin käyttökohteita ovat esimerkiksi tekstipuskurien käsittely ja Vimiin sisältyvän ohjelmointikielen lauseiden suorittaminen. Kaiken kaikkiaan komentorivi on paikka, josta voi vaikuttaa ohjelman toimintaan kaikkein eniten.

Vim script

Vakavasti otettavissa tehotyökaluissa pitää tietysti olla oma ohjelmointikieli – niin on Vimissäkin. Vimin ohjelmointikieli on nimeltään Vim script,



Liike- ja muokkauskomentoja.



ja se muistuttaa hieman Pythonia ja Bourne Shell (sh) -komentotulkin kieltä.

Ohjelmointikielen avulla tekstieditoria voi "skriptata" eli niputtaa useita komentoja tai kokonaisuuksia yhden komennon taakse. Vimissä on tähän tarkoitukseen myös näppäilymakrojen tallennus ja toisto, mutta ohjelmointikieli mahdollistaa monimutkaisemman toimintalogiikan.

Listauksessa 2 on pieni esimerkki-ohjelma, joka luo Vimiin uuden `:Shell`-komennon. Komennolle annetut argumentit suoritetaan käyttöjärjestelmän komentotulkissa (esim. Bash) ja tulos näytetään uudessa Vim-ikkunassa ja puskurissa.

Ohjelman rivillä 1 määritellään funktio nimeltä `shellkomento`. Rivillä 2-3 luodaan uusi ikkuna ja puskur sekä määritellään puskurille muutama asetus, jotka tekevät siitä väliaikaisen. Varsinainen komentotulkin komento suoritetaan rivillä 4. Siinä hyödynnetään Vimin `read`-komentoa, joka ajaa ulkoisen ohjelman ja kirjoittaa tulosteen Vim-ikkunasiin. Funktion lopuksi (rivi 5) siirretään kursori takaisin puskurin ensimmäiselle riville. Rivillä 8 luodaan vielä Vim-ikonen toimiva komento `:Shell`, joka hyväksyy yhden tai useampia argumentteja ja joka kutsuu edellä määriteltyä funktiota.

Omat Vim script -ohjelmat voi tallentaa `~/.vimrc`-tiedostoon, tai ne voi sijoittaa myös `~/.vim/plugin`-hakemistoon, josta ne ladataan automaattisesti Vim-ikäynnistyessä.

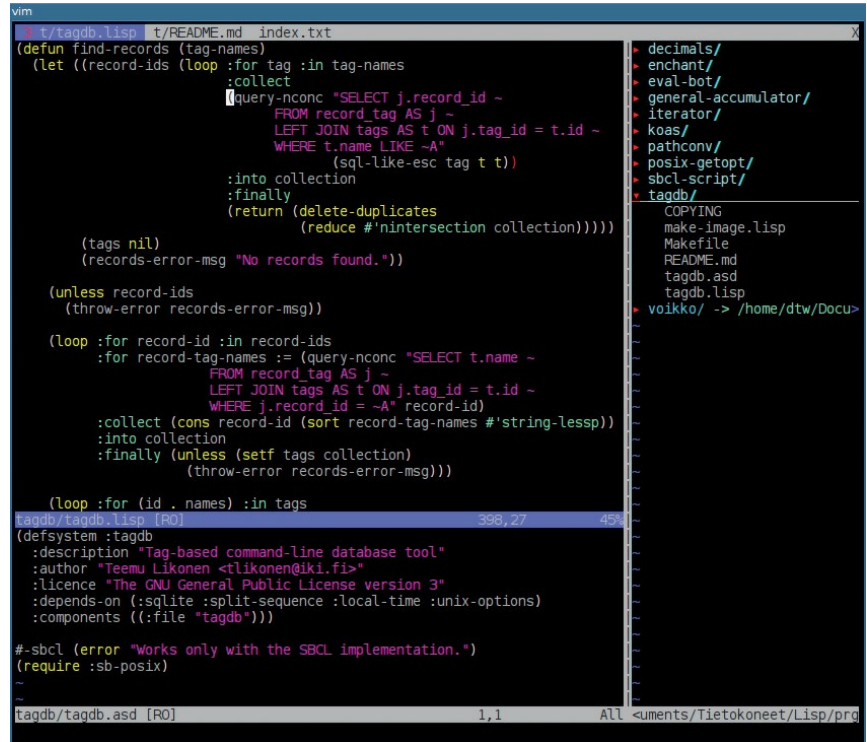
Vim script ei ole yhtä monipuolinen kuin muut nykyajan skriptikielien, mutta se kytkeytyy saumattomasti Vim-ikonen komennot ovat samalla Vim script -kielen lauseita. Jos Vim scriptin ominaisuudet loppuvat kesken ja kaipaa ilmaisuvoimaisempaa kieltä, Vimiä voi ohjelmoida myös esimerkiksi Perl-, Python-, Ruby-, Lua- tai Tcl-kielen avulla.

Tekstipäätteessä ja työpöydällä

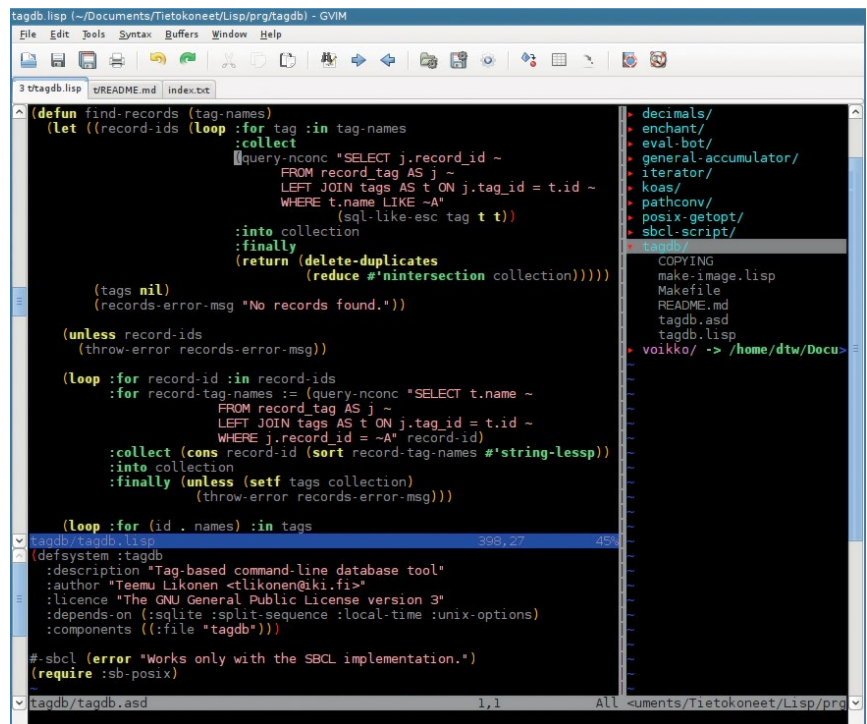
Alkuperäistä Vi:tä käytettiin tekstipäätteyyhteyden kautta, ja se olikin ainoa vaihtoehto. Myös nykyään on paljon tilanteita, joissa graafista käyttöliittymää ei voi käyttää. Esimerkiksi tietokonejärjestelmien ylläpitä-

```
1 function! s:shellkomento(komentorivi)
2   botright new
3   setlocal bufstype=nofile bufhidden=wipe nobuflisted noswapfile nowrap
4   execute 'silent $read !' . a:komentorivi
5   1
6 endfunction
7
8 command! -nargs=+ -complete=file Shell call s:shellkomento(<q-args>)
```

Listaus 2. Oma Vim-komento `:Shell`, joka ajaa komentotulkissa ohjelman ja näyttää tulosteen editorin puskurissa.



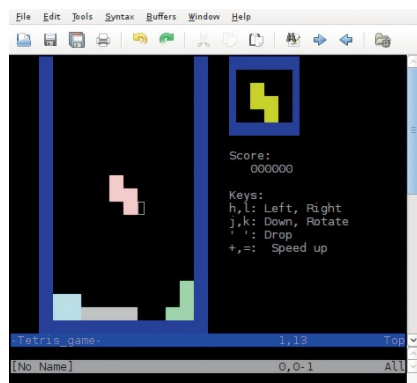
Vim 7.4 tekstipäätteessä (Xterm)...



...ja graafisena.

jät työskentelevät usein etäyhteyden (ssh) varassa ja hallitsevat esimerkiksi yrityksen palvelinohjelmistoa. Silloin tekstipäätteessä toimiva editori on ahkerassa käytössä.

Nykyajan Vimissä on graafinen käyttöliittymä, jossa on hiiren avulla käytettäviä valikoita, työkalupainikkeita sekä muita graafisista ohjelmista tuttuja toimintoja. Vim on kuitenkin suunniteltu sellaiseksi, että kaikki editorin toiminnot ovat käytettävissä tekstipäätteessä ja pelkän näppäimistön avulla. Tehokäyttäjät yleensä suosivatkin näppäinkomentoja, koska niiden käyttö on nopeampaa.



Tetris – se kaikkein tärkein.

Vi muualla

Vi-editorin kahteen toimintatilaan ja yksinkertaisiin näppäinkomentoihin perustuva tekstinmuokkaus on osoittautunut niin suosituksi ja tehokkaaksi, että sama idea on kopioitu toisiinkin ohjelmiin.

Esimerkiksi GNU/Linux-järjestelmien komentotulkit (mm. Bash) ja monet muutkin ohjelmat sisältävät komentorivin tai tekstinsyöttökenttiä, jotka toteutetaan GNU Readline -nimisen ohjelmakirjaston avulla. Sen saa Vi-tilaan, kun lisää tiedostoon `~/inputrc` seuraavan rivin:

```
set editing-mode vi
```

Oletuksena on päällä lisäytila, josta pääsee normaalitilaan tutulla Esc-näppäimellä. Edellä mainitun asetustiedoston avulla voi vaikuttaa paljon muuhunkin, esimerkiksi näppäinkomentoihin. Ohjeita GNU Readlinen toimintoihin saa kirjoittamalla komentotulkissa "man readline".

Vi:n tai Vimin komennot saa moniin editoreihin ja ohjelmien kehitysympäristöihin. Vi-näppäimiä tukevia ohjelmistoja ovat esimerkiksi Eclipse, IntelliJ, Sublime Text ja GNU Emacs. Vi-näppäimillä ja -komendoilla voi ohjata myös Firefox-nettiselainta, kun asentaa siihen Vimperator- tai Pentadactyl-lisäosan. Vastaava Chrome-selaimen lisäosa on nimeltään Vimium.

kin suunniteltu sellaiseksi, että kaikki editorin toiminnot ovat käytettävissä tekstipäätteessä ja pelkän näppäimistön avulla. Tehokäyttäjät yleensä suosivatkin näppäinkomentoja, koska niiden käyttö on nopeampaa.

Miksi Vim?

Maailma on täynnä tekstieditoreja; miksi pitäisi käyttää vanhanaikaista Vimii? Ehkä siksi, että se on todennäköisesti paljon tehokkaampi kuin mikään aiemmin käyttämäsi. Tehoeditorin opettelu kannattaa, jos työskentelee paljon tekstitiedostojen parissa. Lisäksi Vi tai Vim on valmiiksi asennettuna kaikkiiin Unix-tyyppisiin käyttöjärjestelmiin, joten sen hallintaa voi pitää Unix-maailman kansalaistaitona.

Vaikka Vimin varsinainen tarkoitus on olla vain tehokas tekstieditori, siihen sisältyvä ohjelmointikieli tekee mahdolliseksi kaikenlaisten lisäominaisuuksien toteuttamisen. Eri tiedostotyypeille, esimerkiksi lähdekooditiedostoille, on tarjolla vähintäänkin tekstin värjääminen syntaksin mukaisesti. Vakiovarusteisiin kuuluu myös ohjelmakoodin rivien oikeanlainen sientäminen, mutta muitakin ohjelmointia helpottavia toimintoja on saatavilla.

Jotkut viettävät Vimin ja tekstitiedostojen parissa niin paljon aikaa, että mielellään laajentavat editoria perinteisen käyttöalueen ulkopuolelle. On olemassa esimerkiksi lisäosa tiedostojen käsittelyyn, versionhallintaohjelmien kuten Gitin käyttöön sekä muistiinpanojen hallintaan. Pakollinen Tetriskin on tietysti saatavilla.

Vim ei ehkä ole aloittelijan eikä satunnaisen peruskäyttäjän editori, mutta se on saavuttanut suuren suosion tietokoneharrastajien keskuudessa. Sen ympärille on muodostunut käyttäjäjyhteisö, joka tapaa irc-kanavilla ja sähköpostilistoilla. Käyttäjät auttavat toisiaan editointipulmissa, ja osa heistä osallistuu editorin tai sen lisäosien kehittämiseen.

Editorin monipuolinen komentovalikoima on innostanut harrastajia myös leikittelemään. Esimerkiksi netissä Vimgolf.com-sivustolla kilpaillaan, kuka suoriutuu annetuista muokkaustehtävistä pienimmällä näppäinpainallusten määrällä. Lyhyiden ja eleganttien ratkaisujen keksiminen on mukava älyllinen haaste.



– seuraava sukupolvi?

Osa Vim-editorin aktiivikäyttäjistä haluaa kehittää editoria hieman eri suuntaan kuin Vimin pääkehittäjä Bram Mooleenaar, joka puolestaan suhtautuu editorin ominaisuuksiin ja muutoksiin melko konservatiivisesti. Erilaiset tavoitteet johtivat siihen, että Thiago de Arruda perusti Neovim-projektin, joka pyrkii luomaan editorista uudistetun, seuraavan sukupolven version.

Neovim-editori on ulkoisesti pitkälti samanlainen, mutta uudistuksia on luovassa ohjelman sisäiseen toimintaan. Yksi suurimmista tavoitteista on, että editori tukee paremmin rinnakkaisajoa. Esimerkiksi editorin lisäosat toimivat omina prosesseinaan, jotka kommunikoivat Neovim-prosessin kanssa. Erilliset prosessit ja rinnakkainen toiminta antavat lisäosille enemmän vapautta ja mahdollistavat monipuolisempien toimintojen tekemisen. Lisäosat voi kirjoittaa millä ohjelmointikielellä tahansa.

Tarkoituksena on myös erottaa editorin ydintoiminnot ja käyttöliittymä paremmin toisistaan. Tässäkin on kyse sisäisestä muutoksesta, jossa käyttöliittymä toimii melko itsenäisesti ja hyödyntää editorin ytimen palveluita. Rakenne etuna on se, että on helpompaa luoda Neovimille erilaisia käyttöliittymiä ja kytkeä editori osaksi muita ohjelmia.

Projekti on vasta puolentoista vuoden ikäinen, eikä editorista ole vielä virallista julkaisua olemassa. Lähdekoodi ja valmiita asennuspaketteja on kuitenkin saatavilla, ja jotkut ovat jo siirtyneet käyttämään sitä. Lisätietoja saa ohjelman kotisivulta: <http://neovim.io/>.

Aloittelijan ei ehkä kannata lähteä golf-turnaukseen, vaan ensimmäiset Vim-askleet voisi ottaa vaikkapa editorin oman opetusohjelman seurassa. Mukana tuleva vimtutor-ohjelma tuutoroi aloittelijan mukaan tehoeditorin maailmaan. Alussa kannattaa hyödyntää ahkerasti `:help`-komentoa ja sen takana olevia erinomaisia ohjeita. Siitä se lähtee. `:wq` 🍷

Vi-editorien historia

Ed, Em, En, Ex, Vi

Vi-editorin taustalla piilee vanha Unixin rivieditori nimeltä Ed. Nimensä mukaisesti rivieditorit käsittelevät tekstitiedostoja riveittäin. Ohjelman komennoilla käyttäjä saattoi tulostaa näkyviin osan tekstitiedostosta, poistaa rivejä tai kirjoittaa uusia rivejä entisten lisäksi tai tilalle. Rivieditorit olivat kovassa käytössä etenkin silloin, kun tietokoneen ”näyttönä” oli pelkkä tulostin tai kun pääteyhteys mahdollisti vain rivien tulostamisen edellisten perään.

Vähitellen päätteet kehittyivät ja alkoivat osata ohjauskoodeja, joiden avulla voitiin hallita koko kuvaruudun sisältöä. Muutoksen myötä tuli mahdolliseksi entistä visuaalisempien käyttöliittymien toteuttaminen.

Herra nimeltä Bill Joy aloitti opiskelun Kalifornian yliopistossa Berkeleyssä vuonna 1975. Sitten hän työskenteli laitoksella Chuck HALEYN kanssa eräässä ohjelmointiprojektissa, jossa he käyttivät Ed-editoria. Alkeellinen ja hankala tekstieditori aiheutti harmaita hiuksia, mutta lievää helpotusta toi Em-editori (”Editor for Mortals”), jonka oli kehittänyt George Coulouris Lontoon Queen Mary Collegesta. Em ei ollut kokoruutueditori, mutta se hyödynsi joitakin päätteen ohjauskoodeja, ja kursoria pystyi sentään liikuttamaan edestakaisin yksittäisellä rivillä.

Bill Joy paranteli Ed-editoria ja lisäsi siihen Em-editorin ominaisuuksia. Hän kutsui tekelettään ensin nimellä En, myöhemmin nimellä Ex. Uudessa editorissa oli visuaalinen toimintatila, eli siitä oli kehittynyt oikea kokoruutueditori. Editorin nimeksi vakiintui sittemmin Vi, koska yliopiston omassa Unixissa (BSD) oli vi-komento, joka käynnisti Ex-editorin suoraan kokoruututilassa.

Vi-editorin luoja Bill Joy ei itse jatkanut editorin kehitystä kovin pitkään. Sen sijaan kaupallisten Unixien kehittäjät ottivat editorin omakseen 80-luvun alussa ja parantelivat sitä. Vi:stä tuli erottamaton osa Unixia.

Kloonien hyökkäys

Vi-editorin jakelulisenssi ei sallinut ohjelman eikä sen lähdekoodin vapaata levittämistä, joten Vi:tä ei ollut heti saatavilla muihin kuin kaupallisiin Unixeihin. Editorin maine oli kuitenkin kiirinyt jo

kauemmas, ja editori herätti kiinnostusta ja kateutta Unix-maailman ulkopuolella.

1980-luvun lopulla alkoi erilaisten Vi-kloonien vyöry. Tim Thompson julkaisi oman Stevie-nimisen klooninsa Atari ST -tietokoneelle vuonna 1987. Parin vuoden päästä Steve Kirkendall julkaisi Elvis-nimisen kloonin, joka oli aluksi tarkoitettu Minix-käyttöjärjestelmälle mutta levisi sitten muihinkin Unix-tyyppisiin järjestelmiin.

Kloonit eivät olleet täysin samanlaisia kuin Unixin Vi-editorit: niistä saattoi puuttua ominaisuuksia, tai niihin oli lisätty jotakin uutta. BSD-Unixin parissa työskentelevä Keith Bostic kuitenkin halusi täysin samanlaisen, vapaalla jakelulisenssillä varustetun kloonin, jossa myös alkuperäisen editorin bugit on säilytetty. Bosticin ohjelmoima Nvi-niminen klooni julkaistiin 4.BSD-Lite1:n mukana vuonna 1994. Nvi on hyvin merkittävä vapaalla lisenssillä julkaistu Vi-klooni, ja se on edelleen saatavilla nykyajan Linuxeihin ja Unixeihin – joissakin se on jopa oletus-Vi-editorina.

Vi Improved

Kloonituotantoon lähti mukaan myös Bram Moolenaar, joka käytti Amiga-tietokonetta ja halusi sillekin kunnollisen Vi-editorin. Joitakin kloonieja oli Amigalle jo olemassa, mutta Moolenaar ei ollut niihin tyytyväinen. Hän lähti liikkeelle Stevie-editorin lähdekoodista ja ryhtyi parantelemaan sitä. Editorin koodi muuttui niin paljon, että hän päätyi antamaan sille uuden nimen: Vim. Se julkaistiin ensimmäisen kerran Amigan Fish-levykekoelmassa vuonna 1991.

Bram Moolenaarin Vim eli Vi Improved herätti heti kiinnostusta, ja muut ohjelmoijat alkoivat lähettää tekijälle parannusehdotuksia ja koodinpätkiä. Vimin suosio kasvoi, ja se siirrettiin useille käyttöjärjestelmille.

Vuosien aikana Vim on saavuttanut aseman tärkeimpänä ja kehittyneimpänä Vi-editorina. Unix-tyyppisten käyttöjärjestelmien peruskalustoon ja POSIX-standardiin kuuluu Vi-editori, mutta käytännössä sen tilalla on useissa järjestelmissä Vim.

Vimin kehitystä johtaa edelleen sama mies, Bram Moolenaar, mutta kehitykseen ovat osallistuneet vuosien aikana monet muutkin. Oheiseen taulukkoon on koottu tärkeimpiä kehitysvaiheita.

Versio	Julkaisu	Ominaisuuksia
1.14	2.11.1991	Ensimmäinen julkaisu (Amiga, Fish-levyke #591).
1.22	1992	Siirretty Unixille.
3.0	12.8.1994	Tuki useille tekstipuskureille ja ikkunoille.
4.0	29.5.1996	Graafinen käyttöliittymä.
5.0	19.2.1998	Tekstin värit syntaksin perusteella.
6.0	26.9.2001	Tekstialueiden piilottaminen, ohjelmoitavat lisäosat.
7.0	8.5.2006	Oikoluku, välilehdet, älykäs kirjoituksen täydennys, undo-haarat.
7.4	10.8.2013	Uusin versio.

Vim-editorin kehitysvaiheita.



ADM-3A-pääte, Bill Joyn käyttämä Vi-editorin kehitys- ja käyttöympäristö vuonna 1976. Kuva: Chris Jacobs (Wikimedia Commons).



Maasta Kuuhun – ja ehkä takaisin

Ensimmäinen kuulento laukaistiin Cape Canaveralista 16.7.1969. Eagle laskeutui Kuun pinnalle 24.7. Seurasi pieni askel ihmiselle, suuri ihmiskunnalle. Ikiaikainen unelma oli saavutettu. Se on inspiroinut myös pelintekijöitä. Ei tosin monta.

Teksti: Jukka O. Kauppinen

Kuvat: Jukka O. Kauppinen, Wikimedia Commons

Kuu, tuo läheinen ystävämmme. Niin lähellä, niin kaukana, niin vaikeasti saavutettavissa. Tarvittiin kylmä sota, yksi tuhoittoman rikkaan demokratian asevarasto ja niin komea presidentti, että hän pystyi pokkana päättämään, että nyt viedään muutama ukko kä-

velyllä elottomalle kivelle. Avaruuskilpailussa tähän saakka turpiinsa ottanut Yhdysvallat tarttui haasteeseen, ja Apollo 11 laskeutui Kuuhun.

Avaruuskilpajuoksu on kuitenkin inspiroinut yllättävän harvoja pelejä. Ihme ja kumma, sillä aihehan on mitä lumoavin. Lauotaan avaruuteen tyyppäjä ja toivotaan, että lennä Juri Gaga-

rin, tule elävänä takaisin.

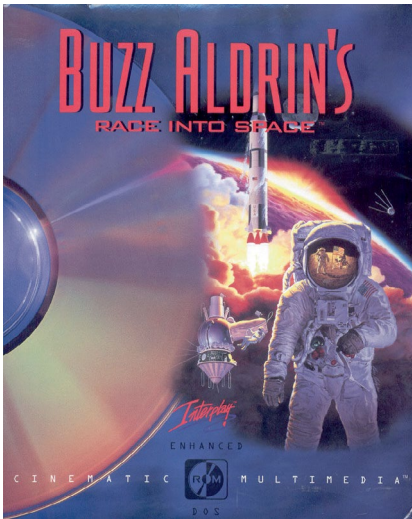
Onneksi jotkut ovat sentään vähän yrittäneet. Tai joku. Se ensimmäinen tyyppi Kuussahan oli Neil Armstrong ja Kuuta kiertävään komentomoduliin jäi isännäksi Michael Collins. Ja sitten se yksi hemmo, se toinen mies Kuussa. Buzz Aldrin. Buzzilla on bisnesvaistoa, ja hän on lainannut nimensä kahteen aivan erityisen hienoon peliin, jotka molemmat kuvaavat Yhdysvaltain ja Neuvostoliiton avaruuskilpajuoksua 1950-luvulta eteenpäin. Ne pelit ovat *Buzz Aldrin's Race Into Space* ja *Buzz Aldrin's Space Program Manager: Race to the Moon*.

FMV-aikakauden kaunis helmi

Buzz Aldrin's Race Into Space (1993, Interplay) on todellinen historiallinen helmi. Se syntyi tähdenlennoksi jääneen Strategic Visions -studion käsissä aikana, jolloin PC-pelit olivat suunnattoman murroksen ääressä. Vanhat 16-bittiset pelikoneet olivat väistymässä PC-juggernautin tieltä, ja CD-ROM



Ensimmäiset kosmonautit palasivat maan pinnalle heittoistuimen avulla.



Buzz Aldrin's Race Into Space, 1993.

teki tuloaan. Vaikka CD-versio oli pelattavuudeltaan identtinen korppupainokseen nähden, oli hopeakiekkopelissä puhutut ääni- ja radioviestinnät sekä tuhattomasti ihan aitoa ja alkupe- räistä NASAn videokuva. Tämä teki jälkimmäisestä hienon elämyksen niille, jotka nauttivat avaruusmatkailun syvistä saloista täydellä sydämellään.

Pelinä Buzzin Race oli teos, joka lämmitti suunnattomasti meikäläisen scifistisydäntä. Se toi kilpajuoksun Kuuhun peliruuduille strategiapeliksi, jossa saattoi valita ohjattavakseen joko Yhdysvaltain tai Neuvostoliiton avaruushjelman. Tästä eteenpäin edettiin askel askeleelta ja kuukausi kuukaudelta kohti korkeuksia, kun pelaajan komentamat tiedemiehet kehittivät avaruustekniikkaa, raketteja ja aluksia, laukoen alkuun miehittämättömiä satelliitteja ja koealuksia, sitten ladaten kapselit koekaniineillaan. Et- hän räjähdä (uudestaan)?

Buzzin molemmista peleistä voidaan itse asiassa puhua samassa yhteydessä, sillä sekä Buzz Aldrin's Race Into Space että Buzz Aldrin's Space Program Manager: Race to the Moon (2014, Matrix Games) pohjautuvat samaan vuonna 1989 julkaistuun Liftoff-lautape- liin. Molemmissa on sama perusmekaniikka, eroja on vain toteutuksessa ja viimeistelyssä.

Interplay oli näet aikansa isoja peli- taloja, jonka teokset olivat AAA-tasoi- sesti hiottuja, ison budjetin laatupele- jä. Matrix Gamesin avaruusstrategiaa sen sijaan on pakko pitää nykyajan melkein-indienä. Tämä näkyy muun muassa siinä, että visuaaliset raamit ja viimeistely ovat vaatimattomampia,



2014 julkaistu Buzz Aldrin's Space Program Manager: Race to the Moon on vasta alkua. Sarjan seuraavassa pelissä katseet ovat jo Marsissa.

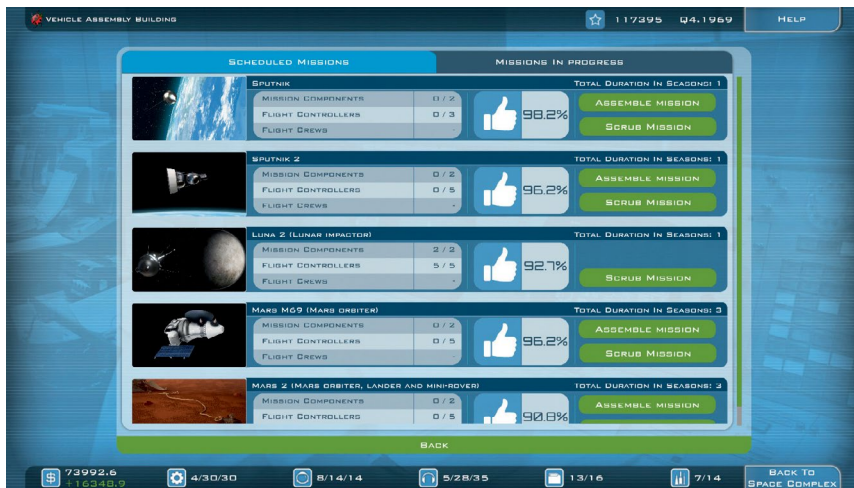
vaikka uudempi peli on myös monel- la tapaa miellyttävämpi ja helpompi pelata sekä paikoittain tyylikkäämpi. Matrixin versiosta on myös nipistetty pois osa lautapelin sisällöstä, joka Interplayn pelistä löytyi.

Sitä ei voi silti kiistää, etteivätkö ne kumpikin tulkitse aihetta upeasti.

Kohti Kuuta, (melkein) hinnalla millä hyvänsä. Palkataan tiedemiehiä kehittä- mään tekniikkaa, raketti-inssejä alati isompien rakettien pariin ja lisää inssi- nautteja satelliittien, avaruuspukujen, miehitettyjen ja miehittämättömien avaruusalusten pariin. Raivaustrak- torit laajentavat laukaisualustoja yhä



Race Into Space oli ja on erinomainen simulaatio suurvaltojen kilpajuoksusta avaruuteen.



Avaruudessa voi kilpajuosta monella rintamalla. Satelliitteja, taivasmiehiä, kuu- ja planeetta-luotaimia. Mihin panostat?

suurempia taivaskieheitä varten ja sekä astronautit että tiedemiehet istuvat välillä koulunpenkillä opettelemassa uusia taitoja.

Vaihtoehtohistoriaa

Jos pelkkä avaruuden tavoittelun lumo ei kiinnosta, pelit tarjoavat myös useita vaihtoehtoisia tutkimussuuntia. Sekä Neuvostoliitolla että Yhdysvalloilla oli piirustuspöydällä useita vaihtoehtoja miehitettyjä avaruus- ja kuulentoja varten, joista molemmat valitsivat yhden pääohjelmaansa varten. Mutta mikään ei estä pelaajaa valitsemasta toisin. Mitä jos -avaruusfantasia onkin ihastuttavan vahvaa.

Tämä tee oma avaruusohjelmasi -valinnanvapaus tuo historiallisesti erittäin tarkkaan avaruusohjelmasimulaatioon upeaa hiekkalaatikon henkeä, vaikka loppumetreillä valinnat tehdäänkin valmiiksi ladotuista kehityspoluista. Samalla jännittävää on sekin, että videopelisisimulaation kautta todellisen historian riskit ja haasteet tuntuvat kaamean valtavilta. Kylmän sodan avaruuskisassa moni virstanpylväs putosi Neuvostoliiton syliin vain viikkojen ja kuukausien kaulalla, mutta kun pelaaja yrittää laukoa rakettaan historialliseen tahtiin, niin johan riittää tulipatsaita laukaisualustoilla. Vainajatkin ovat valmiiksi tuhkattuja.

Liekö koskaan ihmiskunnan historiassa pistetty niin paljoa rahaa niin rai-vokkaan nopeasti kehitettyyn teknologiaan, jota ammuttiin taivaalle niin valtavien riskien saattamina? Välillä molemmilla osapuolilla kaikki meni pieleen, mutta onneksi jenkeillä oli yksi verraton valttikortti – nimittäin

vanha kunnon Werner ”tähtäsin Kuu-hun, osuin Lontooseen” von Braun. Von Braun tiesi mitä tahtoi ja tiesi mitä teki, ja herran ”oho, tein ylimittaisen V2:n” Saturn V:t olivatkin luotettavuudessaan ainutkertaisen hienoja kappineita.

Mutta videopelaajalla on mahdollisuus tyriä Wernerinkin viritykset. Ja se on osa avaruuskisan lumoa. Miten kauan raaskit tai uskallat kehittää tekniikkaa, ennen kuin sytytät tulilangan?

Ilmaiseksi avaruuteen

Avaruustutkimusta voi harjoittaa myös ilmaiseksi. Interplayn Buzz Aldrin’s Race Into Spacen oikeudet palautuivat alkuperäisille kehittäjille vuonna 2005, jolloin nämä aloittivat kuusivuotisen projektin pelin saamiseksi nykyajan tietokoneille. Versio 1.0 Race Into Space – U.S.-Soviet Space Race Simulationista julkaistiin 2011.

Race Into Space on käytännössä hieman hiottu versio lähes kahden vuosikymmenen takaisesta avaruussimulaatiosta. Se on yhä erittäin kiinnostava, haastava ja jännittävä avaruussimulaatio, joskin ei ihan kaikilta osiltaan viimeisen päälle informatiivinen tai pelaajaystävällinen. Käyttöliittymä-



Vuoden 1993 Interplay-peli julkaistiin uudelleen modernisoituna, nykykoneilla toimivana ilmaispeleinä.

Onnistunut tehtävä tuo mainetta ja ehkä lisää budjettiakin; epäonnistuneen lennon jälkeen sankarivainajat herättävät päinvastaista huomiota. Niin, maine on loppumetreillä iso osa minkä tahansa viraston toimintaa, nautittiin sitten kansan rivien tai poliitbyroon luottamusta.

Avaruusohjelmasimulaatioina sekä Race into Space että Space Program Manager ovat hienoja ja ainutlaatuisia pelejä, jotka avaavat avaruuslentojen historiaan monta uutta näkökulmaa. Space Program Managerissa on kuitenkin se bonus, että uudempina ohjelmana sitä tuetaan ja päivitetään yhä, ja jatko-osiakin on luvassa. Sitä voi myös pelata hiekkalaatikossa, jossa avaruuslentoja voi jatkaa myös sen jälkeen, kun ensimmäinen ihminen on saatu Kuun pinnalle. Onhan se vähän päätöntä puuhaa, mutta kuitenkin yksi tapa pelata.

Space Programiin on tulossa myös uusia osia, joista seuraavassa katsotaan jo Mars-planeetan ja avaruussiirtokuntien suuntaan. Ehkä myös avaruusasemat tulevat mukaan kuvaan.

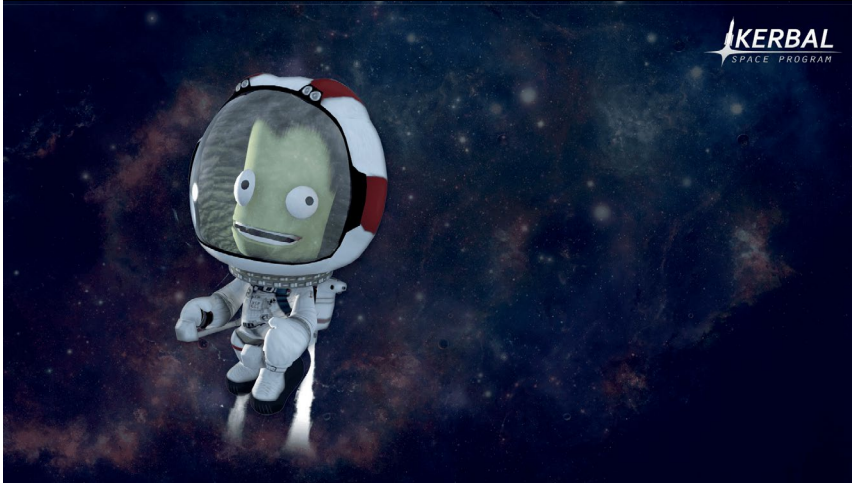
Elämme hienoja avaruusaikoja. Sen kunniaksi: katse tähtiin!

suunnittelu kun on kehittynyt ajan mittaan. Mutta hurjan hyvä peli kaikille aiheesta kiinnostuneille – ja ilmaiseksi.

Hyvin erilainen avaruussimulaatio on Eagle Lander 3D, jo viime vuosikymmenellä julkaistu avaruuslanssimulaatio Eagle-kuumodulin laskeutumisesta arvataksille taivaankappaleelle. Useiden eri Apollo-lentojen laskeutumisosuuden lisäksi simulaatio sisältää myös Apollo-komentoduulin sekä mahdollisuuden kävellä Kuun pinnalla. Ilmainen share ware -versio sisältää pelkän laskeutumisosuuden, mutta täysversion ostamalla voi leikkiä jo manipulisemmin..



Onko sinusta kuunautiksi? Se selviää Eagle Lander 3D:ssä, joskin siinä voi aina yrittää uudelleen.



Kerbaalien avaruusretki

Äitini kommentoi joskus ivallisesti, ettei tietokonepelien pelaaminen ole mitään rakettitiedettä. Väärässä olit, mutsi! Kerbal Space Program korvaa historiallisuuden vapaudella, mutta säilyttää oikeasta avaruusmatkailusta tieteen, vaikeuden, seikkailun ja saavutusten riemun. Tosin nyt saa myös räjähtää ja nauttia ilotulituksesta. KABOOM!

Teksti: Jani Salomaa

Kerbal Space Programin peruspelimoodi on hiekkalaatikko, joka antaa valtavan kasan realistisen tuntuisia osia, joista voi kasata avaruusroketteja ja -lentokoneita oman mielensä mukaan. Riittävänä haasteena on omaamme mukaileva aurinkokunta. Realismi näkyy siinä, että jos kopioi Saturn V -raketin designin, niin sillä pääsee kuuhun asti, jos lentää oikein.

Vaan ihan ensiksi pitäisi päästä Maata muistuttavan Kerbinin kamaralta edes ilmaan. Siihen vaaditaan söpö kerbonautilio nimeltä Jebadiah, komentomoduuli, moottori ja polttoainetta. Helppoa kuin heinänteko. Paitsi, että unohdin laskuvarjon. Hupsis! Takaisin piirustuspöydälle parissa sekunnissa, siis.

Nälkä kasvaa kauemmaksi lentäessä, mutta vaikeusaste nousee samalla. Mahdollisuuksia on pienen mielikuvituksen verran. Saat vaikka rakentaa omia avaruusasemia tai vaikka val-

tavalla lämpökilvellä suojatun aurinkoon syöksyvän vempeleen. Jos kaipaa pelilleen rakennetta, ura- ja tiedemoo- deissa edetään tehtävä tehtävältä korkeammalle.

Kerbal-oliot pehmentävät pelikokemusta. Ne kuolevat törmäyksissä ja räjähdyksissä, mutta eivät menehdy hapen puutteeseen. Hyvä niin, sillä parinkymmenen katastrofin opettamana Jebadiah pääsi lopulta vakaalle kiertoradalle. Karvat pystyssä katsoin ylpeänä auringonnousua. Päihitin sen! Lähin kuu, Mun, näyttää olevan kutsuvan lähellä... Ja sitten tajusin, ettei tankissa ollut polttoainetta maan kamaralle palaamiseen. Hupsis! Nauti maisemista, Jebadiah. Pian opin pelastamaan sinut kiertoradalta.

Alusten kasaaminen osista on periaatteessa helppoa: ihan kuin laittaisi legoja yhteen mainion käyttöliittymän auttamana. Myös alusten ohjaaminen on varsin vaivatonta. Nämä ovat sitten suunnilleen ainoat helpotukset, sillä muuten peli on armoton, ainakin mitä

fysiikkaan tulee. Raketeissa pitää ottaa huomioon muun muassa kuorman ja polttoaineen massa, rakenteiden kestävyys, painopiste, aerodynamiikka, työntövoima, kantorakettien hylkäys löpön lo-
puttua ja lämpö-

suojaus ilmakehään palatessa. Ja vasta oikeassa avaruudessa toimiminen, lentäminen ja kiertoratojen muuttaminen, on kunnan aivojen sekoittaja.

Parasta KSP:ssä on se, ettei mikään pakota oppimaan, mutta seikkailun mahdollisuus motivoi siihen. Jos haluaa leikkiä hupsuilla virityksillä, niin antaa mennä vaan. Jos haluaa asettaa itselleen tavoitteita, peli venyy vaikka mihin. Ellei veny, siihen löytyy todennäköisesti modi.

Kerbal Space Program on hiekkalaatikko parhaimmillaan. Olet sitten tavallinen tallaja tai astrofysiikko, niin siitä löytyy jokaiselle leikkittävää. Jotain sellaista, joka saa tuntemaan, että olet voittanut itsesi ja avaruuden, et vain peliä.

Kun Jebadiah kaikkien koettelemustensa jälkeen käveli Munin pinnalla ja iski lippuni siihen, itkin ilosta. Oikeasti. Iso mies. Koska se oli kaikki minun työtäni. Minun saavutukseni.

Nostin katseeni ylös ja mietin: mihin seuraavaksi?

Buzz Aldrin's Space Program Manager: Race to the Moon

Julkaisu: 2014 – hinta 29 euroa
Tekijä: Polar Motion / Matrix Games
Saatavilla: Android, iOS, Mac, PC
www.matrixgames.com

Buzz Aldrin's Race into Space

Julkaisu: 1993 – retro
Tekijä: Strategic Visions / Interplay
Saatavilla: PC
www.mobygames.com/game/buzz-aldrins-race-into-space

Race Into Space

Julkaisu: 2011 – ilmainen
Tekijä: Strategic Visions
Saatavilla: Linux, Mac, PC
www.raceintospace.org

Kerbal Space Program

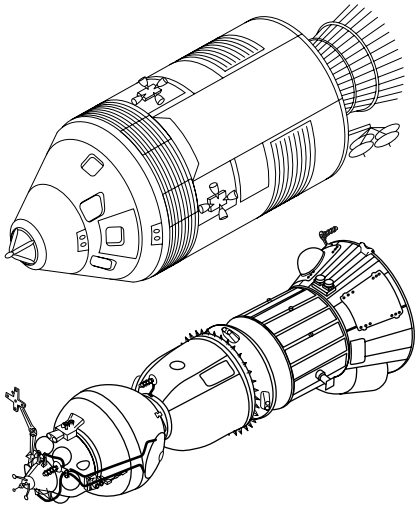
Julkaisu: 2015 – 40 euroa
Tekijä: Squad
Saatavilla: Linux, Mac, PC
www.kerbalspaceprogram.com

Eagle Lander

Julkaisu: 2009 – ilmainen / 25 dollaria
Tekijä: Go For Landing LLC
Saatavilla: PC
www.eaglelander3d.com



Kerbal Space Program on ihastuttava avaruushiekkalaatikko.



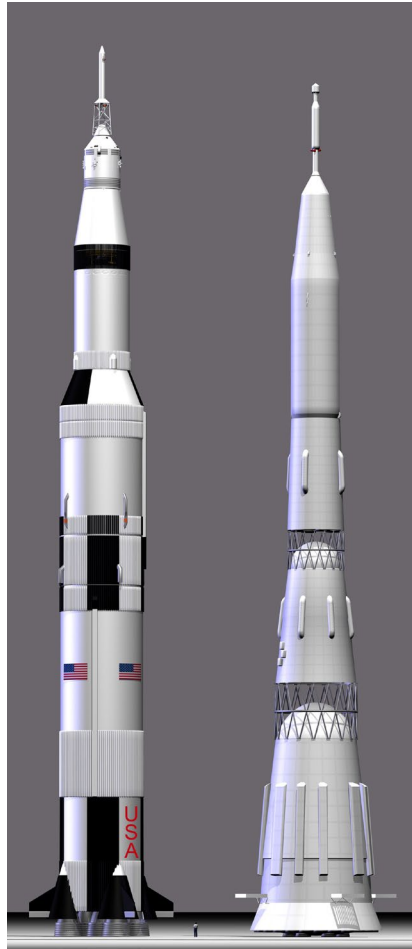
Neuvostoliiton kuuraketti

Yhdysvallat ehti ensin Kuuhun, mutta kyllä Neuvostoliittokin yritti. Punalipun kuurakettiohjelman olomassaolo tosin tunnustettiin vasta Neukkulan romahduksen jälkeen, eikä mikään ihme. Homma nimittäin kusi totaalisesti.

Kuva: Fred the Oyster / Wikimedia Commons

Neuvoštoliiton avaruussaavutukset 1950- ja 1960-luvuilla olivat päättähuimaavia, eikä Venäjän osaamista tällä saralla kannata väheksyä edelleenkään. Kyllä siellä osataan! Mutta aina ei ole osattu. Zond-ohjelman mukainen miehitetty lento Kuun ympäri Soyuz 7K-OK -avaruusaluksella oli määrä laukaista 8.12.1968, mutta se peruttiin, kun alus ei ollut riittävän luotettava edes Neuvostoliiton standardeilla. Apollo 8 nappasikin pokaalin ympärilennosta 21.12.1968. Täpärälle jäi. Kun tappio tuli, neuvostojohto perui ohjelman saman tien.

Entäs kosmonautti Kuussa? N1/L3-ohjelman mukaan Kuuhun oli määrä laukaista etukäteen tukikohta-alus kahden kuumönkijän kera, joita seuraisi muutaman viikon kuluttua miehitetty lento. LK-kuualuksella ehdittiin suorittaa useita miehittämättömiä lentoja Maan kiertoradalla, ja myös Apolloa vastaava 7K-LOK-avaruusalus osoit-



Yhdysvaltojen Saturn V -kantoraketti ei pettänyt kertaakaan. Neuvostoliiton N1 joka kerta.

tautui koelentoilla toimivaksi värkiksi. Neuvostoliiton kuulento-ohjelmassa kuitenkin riitti muita ongelmia, ja sen lopullinen kuolinisku oli N1-kantoraketin täydellinen epäonnistuminen. Rakettihistoria kertoo, että Neuvostoliiton poliittinen johto hoputti laukaisuja, vaikka tiedemiehet tiesivät, etteivät raketit ole luotettavia. N1:n kehitys joutui myös pahoin harhateille, kun Neuvostoliiton rakettinero, ”avaruuslentojen isä” Sergei Korolev, menehtyi kesken projektin. Politbyroon ”ampukaa se raketti tai päitä putoilee” -ukaasien jälkeen Neuvostoliiton avaruusohjelma vajosi historialliseen pohjamutaansa: neljä laukaisua, neljä räjähtänyttä kantorakettia, ei kosmonauttia Kuussa.

Suurvallan arvovalta ei kestänyt nöyryytystä, joten N1-rakettien olemassaoloa, paukkumisia tai edes kuulento-ohjelmaa ei tunnettu ulkomaailmassa virallisesti ennen kuin äimistyneet läntiset tutkijat löysivät niistä haudatut raportit Neuvostoliiton jäämistöarkistoista glasnostin alettua.

Tietokoneella Kuuhun

Teksti: Ville-Matias Heikkilä ja Jukka O. Kauppinen

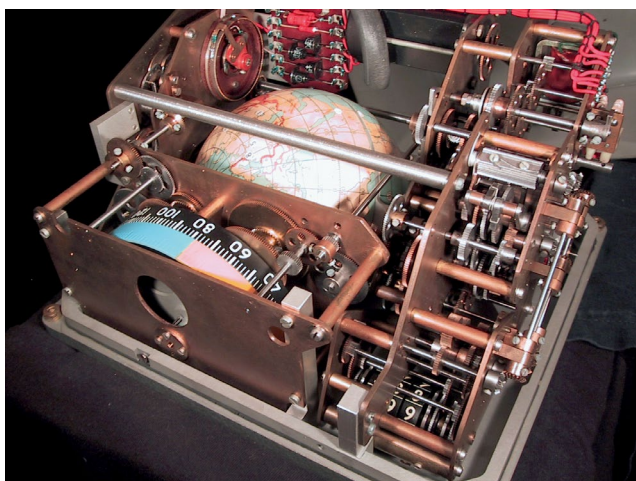
Yhdysvallat oli tietotekniikan pioneeri, ja kuulentoihin käytetyissä Apollo-avaruusaluksissa ja Eagle-kuumoduulissa onkin aikaansa nähden erittäin kehittynyt tietokone. Mutta myös Neuvostoliitossa käsitettiin tietotekniikan tärkeys avaruuslennoissa. Jo Voshod-avaruusaluksissa oli äimistyttävä sähkömekaaninen Globus-tietokone, joka laski neuvostoaluksen sijaintia avaruudessa käsittämättömän monimutkaisella järjestelmällä.

Globus sisälsi satoja mekaanisia osia, jotka pyörittivät laitteen pääkonsolissa pientä maapalloa. Tästä kosmonautit näkivät jatkuvasti, minkä maanosan yllä alus oli. Erittäin tarkka ja luotettava mekaaninen sijainninmääritys oli käytössä Neuvostoliiton ja Venäjän avaruusaluksissa aina vuoteen 2002 saakka.

Neuvostoliitolla kun ei ollut käytössään yhtä kattavaa viestintäasemien verkostoa kuin USA:lla, niin kosmonautit tarvitsivat tarkempaa mukana kulkevaa sijaintitietoa kuin Yhdysvaltain astronautit. Globus oli myös tärkeä osa paluujärjestelmää, jos kosmonauttien täytyi ottaa avaruusalus käsiöjhaukseen.

Kosmonauttikin tarvitsi silti kuulentoa jotain järeämpää, ja neuvostotiedemies otti haasteen vastaan. Zond-kuuohjelman 7K-avaruusaluksiin kehitettiin täysin digitaalinen Argon-11S-tietokone. Äärimmäisiin olosuhteisiin tarkoitettu tietokone kesti 10 G:n kiihtyvyyden ja kaikki sen toiminnot oli varmistettu kolminkertaisesti. Koneen RAM-muisti koostui 128:sta 14-bittisestä sanasta ja ROMia oli 4096 17-bittistä sanaa.

Vaikka kuulentoa ei koskaan toteutettuakaan, ohjasi Argon-11S-tietokone Kuun ympäri kiertäneen miehittämättömän Zond 7 -avaruusaluksen turvalisästi sinne ja takaisin elokuussa 1969. Myöhempää Argon 16 -tietokonetta käytettiin kantorakettien ohjaustietokoneina, Saljut-, Almaz- ja Mir-avaruusasemissa sekä Sojuz- ja Progress-avaruusaluksissa aina 2000-luvulle



Globus oli sähkömekaaninen paikannuslaite, kuin avaruusaluksen GPS.



asti. Satojen lentojen aikana koneessa ilmeni jokunen ohjelmisto-ongelma mutta ei ainuttakaan laitteistovikaa, mitä voidaan pitää uskomattomana saavutuksena.

Myös Yhdysvaltain Apollo-avaruusaluksissa osattiin. Niihin suunniteltu Apollo Guidance Computer (AGC) oli paitsi ensimmäisiä moderneja sulautettuja järjestelmiä, myös ensimmäisiä integroituihin piireihin perustuvia tietokoneita. AGC:n tehoja on monissa jutuissa verrattu taskulaskimiin, mutta todellisuudessa sillä ei ole mitään hävettävää esimerkiksi 80-luvun kotimikrojen rinnalla. Koneen sananleveys oli 16 bittiä, pääkellotaajuus 2 MHz, ja siinä oli RAM-muistia nykytermeissä 4 kilotavua. AGC:n DSKY-käyttöliittymäpaneeli koostui numeronäppäimistöä ja releohjatus-

ta seitsensegmenttinäytöstä. AGC:llä pääsee nykyisin leikkimään myös emulaattoriprojektien kautta.

AGC:n toimintaa ohjasi moniajava reaaliaikakäyttöjärjestelmä, jota säilönyt 72 kilotavun ROM-muisti oli kirjaimellisesti käsityötä: tekstiilialan ammattilaiset nypläsivät yksittäiset bitit paikoilleen pujottamalla johtimia ferriittirenkaiden läpi, minkä vuoksi sitä kutsuttiin myös Little Old Lady- eli LOL-muistiksi.

Ensimmäinen kuulaskeutuminen tuotti yllättävän ongelman – kesken kaiken. 386 000 kilometriä kotoa, vauhdilla vierasta taivaankappaletta kohti syöksyvän avaruusaluksen tietokone näet päätti, että nyt en jaksa, liikaa tekemistä, rebootaan. Tutkalaitteen ja tietokoneen välinen viestiliikenne oli joutunut epäsynkkiin,

mikä aiheutti koneelle valtavan keskeytysvyöryn. Käyttöjärjestelmä ei ehtinyt käsitellä keskeytyksiä, joten se hätäratkaisuna boottasi koneen ja antoi surullisen kuuluisan virhekoodin 1202. Tässä vaiheessa amerikkalaisten avaruuslentäjien taidot pantiin toden teolla puntariin. Tietokone jumissa, nyt mennään käsipelillä – ja kaiken lisäksi ohi maalialueesta.

Ensimmäinen ihmisen avaruusalus Kuuhun lensikin tässä vaiheessa vaakalentoa 122 metrin korkeudessa. Polttoaine hupeni hurjaa vauhtia ja Neil Armstrong ohjasi avaruusaluksaan käsiohjauksella etsien tasaista paikkaa, jonne aluksen voisi istuttaa. Armstrong ohjasi avaruusaluksiaan yhä alemmas ja alemmas, kiitäen lopulta kolmen metrin korkeudella kuupölypilvessä, arvioiden lopulta aluksen korkeutta aluksen ja kuun pinnalla olevien kivien varjon perusteella. Lopulta lentäjä löi päälle jarruraketit, pysäytti aluksen ja vajuutti sen Kuun pinnalle.

”Houston, Tranquility Base here. Eagle has landed.”

Polttoainetta oli jäljellä enää 25 sekunniksi.

Joskus maailman kehittyneinkään tietotekniikka ei riitä, vaan tarvitaan astronautti hoitamaan homma kotiin. Tai siis Kuuhun.

P.S. Skrolli julkaisisi mielellään laajemmankin artikkelin Neuvostoliiton ja USA:n varhaisilla – ja miksei myös myöhemmillä – avaruuslennoilla käytämästä tietotekniikasta. Soita suutasi toimituksen suuntaan, jos koet aiheen omaksesi! 🐘



Zond-avaruusalusten tulevaa kuulentoa varten kehitettiin täysin uusi digitaalinen ja vuosikymmeniä Neuvostoliiton avaruusohjelmaa erittäin luotettavasti palvellut Argon-tietokoneiden perhe. Ebs08, Wikimedia Commons.



Lisää tallennustilaa Amigaan

Onko Amiga kaivettu kaapista ja korppuja vaihdeltu jo niin paljon, että tenniskyynärpäätä kolottaa? On aika siirtyä pohtimaan kiintolevyn tai CF-kortin asennusta. Vanhan tietokoneen ja modernin tallennusteknologian yhdistäminen ei kuitenkaan aina käy aivan helposti.

Teksti: Ville Jouppi

Kuvat: Manu Pärssinen, Ville Jouppi

Jos koneesi on Amiga 600, 1200 tai 4000, siinä on IDE-väylä suoraan emolevyllä. Amiga 3000:n emolevyllä on IDE:n sijaan nopea DMA-SCSI-väylä. Muiden mallien omistajien pitää sovitella paikalleen jokin sisäinen tai ulkoinen levyohjain.

Tässä artikkelissa keskitytään lähinnä emolevylle integroituihin IDE-väyläisiin ohjaimiin ja MC68000-suorittimen kantaan asennettaviin, Amiga 600:n IDE-väylää emuloiviin kortteihin. Käyttöjärjestelmän suhteen oletamme, että käytössä on vähintään Kickstartin ja Workbenchin versio 2.05, revisio 37.300.

Neljän gigan tuolle puolen

Amigan kiintolevyistä puhuttaessa tulee usein vastaan neljän gigatavun raja. Kyseessä on suurin luku, joka voidaan esittää 32-bittisessä muuttujassa. Kun muuttujan lukualue pyörii ympäri, aloitetaan luku- tai kirjoitusoperaation paikan laskeminen levyn alusta uudelleen. Huonolla tuurilla kirjoitetaan jonkin tiedoston sisältö uusiksi, ja todella huonolla tuurilla osiotaulu sotkeentuu. Tällöin kiintolevyajuri ei enää löydä osioita, kun kone käynnistetään seuraavan kerran.

Ongelman ensimmäisen osan muodostaa kiintolevyajuri. Se sisältää 32-bittistä matematiikkaa, joka rajoittaa suurimman osoitettavissa olevan loogisen tietueen neljän gigatavun kohdalle. Toinen pulma on tiedostojärjestelmä. Myös sen täytyy ymmärtää 64-bittisen matematiikan päälle, jotta laskurit eivät pyörähdä ympäri neljän miljardin kohdalla.

Ongelmaksi voi muodostua myös itse sovellusohjelma. Harvempi sovellus kuitenkin haluaa käsitellä kiintolevyä suoraan, vaan tämä jätetään tiedostojärjestelmän huoleksi. Käytännössä sovellustason ongelmia voi tulla erilaisten levynkorjaus-, eheyty- ja varmuuskopiointiohjelmien kanssa. Myös jotkin emulaattorit käsittelevät levyä suoraan, joten niitä käytettäessä täytyy myös pitää huoli siitä, että virtuaalilevykuvat sijaitsevat neljän gigatavun rajan alapuolella.

Valtaosa ohjelmista saadaan siis tukemaan yli neligigaisia kiintolevyjä, kun korjataan kiintolevyohjaimen ajuri ja tiedostojärjestelmä. Rajan ylittäminen onnistuu kolmella tavalla: TD64, NSD ja DirectSCSI.

Kaksi ensimmäistä ovat saman idean eri toteutustapoja, eli alkuperäisten luku- ja kirjoituskäskyjen lisäksi kiin-

tolevyajuriin on lisätty niistä 64-bittiset versiot. TD64 on itsenäisten Amiga-kehittäjien muodostaman porukan kehittämä standardi, ja NSD on erään Amiga Technologiesin työntekijän yksin kehittämä. Tähän luonnollisesti liittyi draamaa, kuten aina Amiga-piireissä. Molemmat tavat kuitenkin toimivat kutakuinkin samalla periaatteella, ja kummassakin tapauksessa sekä kiintolevyajuri että tiedostojärjestelmä pitää vaihtaa tai paikata.

DirectSCSI puolestaan siirtää vastuuta tiedostojärjestelmälle, eli tällöin tiedostojärjestelmä käskyy suorilla SCSI-käskyillä kiintolevyä toimittamaan halutut sektorit. Kaikki SCSI-ohjaimet eivät tue TD64:ää eivätkä NSD:tä, jolloin DirectSCSI on toimiva vaihtoehto. DirectSCSI:n tapauksessa vaihdetaan yleensä pelkkä tiedostojärjestelmä, mutta joissain kiintolevyajureissa on puutteita, jotka silti asettavat rajan esimerkiksi 1 Gt:n tai 7,8 Gt:n kohdalle.

Huomattavaa on, että 8 Gt:n nimelliskapasiteetilla varustettu CompactFlash-kortti mahtuu oikeasti 7,8 Gt:n rajan sisälle, joten pelkkä tiedostojärjestelmän päivitys riittää 8-gigaisen CF-kortin kanssa. Paikkaamatonta jätetty ROMissa asuva A600:n tai

A1200:n scsi.device osaa osoittaa suurimmillaan 16 383:a uraa, 16:ta lukupäätä ja 63:a sektoria per ura. Kun näihin lasketaan mukaan sektorikoko 512 tavua, saadaan todellinen raja: $16\ 383 \times 16 \times 63 \times 512 = 8\ 455\ 200\ 768$ tavua eli 7,8 Gt.

Pääpiirteittäin jako menee niin, että OS 3.5:n ja 3.9:n käyttäjät valitsevat joko NSD:n tai DirectSCSI:n, kun taas 3.1:n ja vanhempien käyttäjä suuntaa katseensa kohti TD64:ää tai DirectSCSI:tä.

Muiden kuin Commodoren toimitamien levyohjainten käyttäjien kannattaa tarkistaa, sattuisiko oma ohjain tukemaan TD64-käskykanta. Aina-kin kaikki Phase5:n SCSI-ohjaimet ovat TD64-yhteensopivia, ja myös useimmat Jens Schönfeldin tuotteet noudattavat TD64-standardia.

Oma suositukseni on ensisijaisesti TD64, mikäli kiintolevyajurista tuki löytyy, ja toisena DirectSCSI, mikäli kiintolevyajuri sattuu tukemaan suoria SCSI-komentoja.

Useimpien oikeiden SCSI-ohjainten tapauksessa DirectSCSI:llä voidaan osoittaa kahteen teratavuun asti. Kuitenkin esimerkiksi A590/A2091:n ajurit versioon 6.6 asti tukevat ainoastaan yhteen gigatavuun asti. Versiolla 7.0 saadaan mukaan laajennetut SCSI-käskyt ja raja nousee kahteen teraan.

Jos ajossa on OS 3.5 tai 3.9, Set-Patch-komento päivittää sisäistä IDE-ohjainta käskyttävän scsi.device:n ja käynnistää koneen uudelleen. Neljän gigatavun raja rikotaan joka kylmäkäynnistyksellä ja uudeksi rajaksi tulee 128 gigatavua. Voit käyttää NSD- tai DirectSCSI-yhteensopivaa tiedostojärjestelmää. Käynnistysosion tulee sijaita neljän gigatavun rajan alapuolella, mikäli valitsemasi tiedostojärjestelmä ei tue sekä NSD:tä että DirectSCSI:tä.

Vanhempien käyttöjärjestelmien käyttäjillä on vaihtoehtoina IDE-Fix tai yksityisten kehittäjien tekemät päivitettyt ajurit. IDE-Fix-paketti (nykyään IDE-Max) päivittää scsi.device:n tukemaan NSD:tä, TD64:ää ja DirectSCSI:tä 128 gigatavuun asti.

Kyseessä on maksullinen tuote, jonka voit ostaa joltain jäljellä olevista Amiga-jälleenmyyjistä.

IDE-Fixin asentamisen jälkeen täytyy korvata s:startup-sequenceen ilmestynyt rivi "c:idefix" rivillä "c:loadide reset", joka tekee IDE-Fixin ajurista resetinkestävän.

Ilmainen ratkaisu aiheuttaa hieman enemmän vaivaa. Joudut hakemaan netistä TD64-yhteensopivan scsi.device:n ja sille päivityksen, joka poistaa beetaversion

aikarajan. Lisäksi tarvitaan LoadModule-käsky, jolla ajuri ladataan muistiin kylmäkäynnistuksen jälkeen.

Kätevä oikotie tiedostojen haalimiseen on hakea ClassicWB 68k -paketti ja kopioida tarvittavat tiedostot System.hdf:stä ulos. Tiedostot löytyvät hakemiston MyFiles/LargeHD alla olevista hakemistoista. Suosittelem ensisijaisesti hakemistosta SCSI_v43_45_ChrisToni löytyvää ajuria, koska LBA48-tuen lisäksi se korjaa yhden muistilyvuodon scsi.devicestä. Mikäli sieltä ei löydy sinun koneellesi

sopivaa, seuraavaksi paras valinta on SCSI_v43_24.

Jos omaa konettasi vastaavan 43.45-ajurin puute jäi kaivelemaan mieltä, version 43.45 paikkatiedostot löytyvät Aminet-tiedostopalvelusta (www.aminet.net). Jos käytössäsi on AmigaOS 3.9, voit itse purkaa AmigaOS ROM Update -tiedostosta konettasi vastaavan scsi.device:n ja paikata sen Aminetistä haetun paketin ohjeiden mukaan.

A500:n 68000-kantaan liitettävät MKIDE ja Kipper2k:n IDE käyttävät samaa ajuria kuin A600. Muitakin ajuriversioita on tarjolla ClassicWB:n mukana, mutta niiden väliset erot eivät ole tässä merkitseviä.

PFS3 AIO -tiedostojärjestelmä ja LoadModule-käsky löytyvät myös samasta hakemistopuusta, joten sinulla on kaikki kätevästi saatavilla. Voit kopioida tiedostot ulos WinUAE:lla, kun lisäät ClassicWB-konfiguraatioon kiintolevyksi jonkin Windowsissa olevan hakemiston. Sitten vain emulaattori käyntiin ja kopioimaan ne emulaattorin sisällä "hakemistokiintolevyllä".

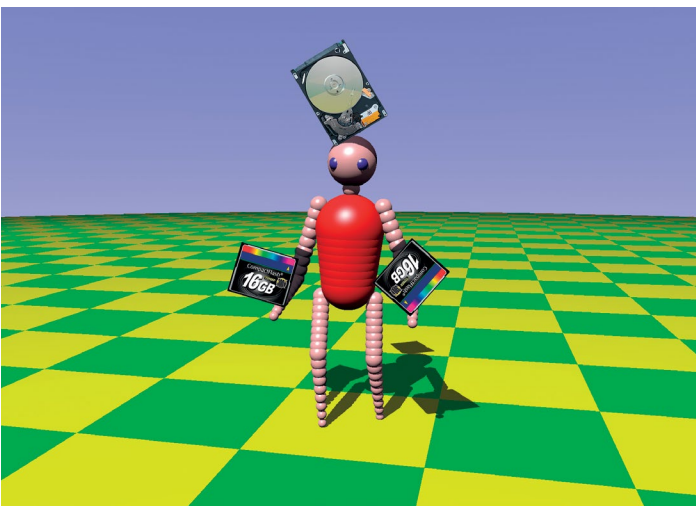
Tiedot järjestykseen

Tiedostoja ei voi heitellä massamuistille hajan hajan, vaan tarvitaan tiedostojärjestelmä pitämään kirjaa niiden nimistä ja sijainnista. Amigalla on käytössä oikeastaan neljä eri tiedostojärjestelmää:

- Muinainen OFS, Old FileSystem, joka oli jo kickstart 1.0:ssa.
- FFS, Fast FileSystem, joka esiteltiin 1.3:n myötä ensin levy pohjaisena, 2.0:sta eteenpäin ROMissa.
- PFS3, Professional FileSystemin kolmas versio, aikanaan kaupallinen, nykyään avointa lähdekoodia.
- SFS, Smart FileSystem, kaksi eri kehityspuuta, ilmainen.

OFS voidaan unohtaa kiintolevyjen kanssa heti kättelyssä, sillä sen suurin osiokoko on jossain 40 megatavun hujakoilla, ja sen sisäinen kirjanpito aiheuttaa paljon lukupään edestakaista liikettä. Valtaosa Amigan korpuista on OFS-formaatissa yhteensopivuuden vuoksi.

FFS oli aikanaan siedettävän nopea, mutta osion maksimikoko on kaksi gigatavua, ja jos kone kaatuu kesken kirjoitusoperaation, seuraavan käynnistuksen yhteydessä on edessä pitkä ja muistia syövä validointi. Sadan me-



Kiintolevy hallintaan

Emolevyllä integroidun IDE-ohjaimen ajuri löytyy suoraan Kickstart-ROMista versiosta 2.05 (37.300) eteenpäin. Historiallisista syistä ajurin nimi on scsi.device, vaikka se jutteleekin IDE-levyille.

Versioita on lukuisia, mutta yhteistä niille kaikille on, että normaalitilanteessa raja menee 4 Gt:n kohdalla. Jos käytetään ROMista löytyvää vakioajuria ja DirectSCSI-yhteensopivaa tiedostojärjestelmää, raja kasvaa 7,8 gigatavuun asti.

gan validointia jaksaa vielä odotella, mutta useamman gigatavun kokoisissa osioissa menee pahimmillaan kymmeniä minutteja. Suurten osioiden kanssa validointi ei välttämättä pääse koskaan loppuun, sillä koneen muisti voi loppua kesken.

PFS3 ja SFS ovat molemmat hollantilaista tuotantoa. Molemmat tarjoavat yli 32 merkin mittaiset tiedoston nimet, atomiset päivitykset ja tuen isoille osioille. Atomisten päivitysten ansiosta OFS:n ja FFS:n validointirumbaa ei tarvitse käydä läpi, jos kone kaatuu tai virrat katkeavat, vaan tiedostojärjestelmä vain unohtaa viimeisimmät muutokset.

PFS3:sta löytyy kolme eri versiota. Normaali PFS3 tukee TD64-käskykanta, PFS3ds on DirectSCSI-yhteensopiva ja myöhemmin kehitetty PFS3aio tukee kaikkia kolmea: TD64:ää, NSD:tä ja DirectSCSI:tä.

SFS:stä löytyy kahta eri pääversiota. Vanha alkuperäisen kehittäjän 1.84-versio tukee kaikkia kolmea standardia ja uudempi 1.279 vain TD64:ää ja NSD:tä. Itse valitsin PFS3aio:n, sillä siinä on hyvä tuki kaikille standardeille, ja se tarkistaa, voiko osion loppua lukea. Jos PFS3aio ei pysty lukemaan osion viimeistä tavua, se ei kiinnitä osiota lainkaan, jolloin tiedon menetyksen riski pienenee huomattavasti.

Kurkistus työkalupakkiin

Valtaosa vartenotettavista Amigan kiintolevyajureista tukee RDB-osiointistandardia. Se on huomattavasti monipuolisempi kuin PC:ssä käytetty MBR-osiointi. Muutamia ominaisuuksia mainitakseni asematunnukset ovat käyttäjän valittavissa ja tulevat RDB:stä. Sinne voi myös asentaa tiedostojärjestelmiä. Käyttäjä voi valita, kiinnitetäänkö osio automaattisesti, mitä tiedostojärjestelmää käytetään ja niin edelleen.

Useimpien kiintolevyohjaimien mukana tulevalla oheiskorpulla on jonkinlainen osiointityökalu, mutta käytännössä RDB-yhteensopivan ohjaimen kanssa on syytä tehdä valinta joko HDToolBoxin tai HDInstToolsin väliltä. Muiden valmistajien osiointityökalut eivät välttämättä tue kaikkia RDB:n ominaisuuksia tai sekoilevat isojen kiintolevyjen kanssa pahasti.

HDToolBox

HDToolBox on Commodoren itsensä kehittämä työkalu. Se esiteltiin A590/2091-ohjainten myötä, ja sillä voi muuttaa kaikkia RDB:n tukemia asetuksia. Versio 2.22 toimii vielä kickstart 1.3:n alla, mutta sen jälkeen tulleet versiot vaativat 2.04:n tai uudemman. Versio 2.22 löytyy hämäävästi Install 2.04, 2.05 ja 2.1 -levyiltä, vaikkei se 2.0:aa vaadikaan. 2.04:n ja uudempien käyttäjille suosittelen 3.1-version Install-levykkeen mukana tulevaa versiota 40.4.

Nämä vanhemmat HDToolBoxin versiot eivät osaa näyttää suuria lukuja, joten tulet luultavasti näkemään liian pieniä ja negatiivisia lukuja, kun osioiden ja levyjen kokoja laskeaan. Geometria-arvot ovat kuitenkin ne, joilla pelataan, ja ne eivät vuoda yli, joten haitta on lähinnä kosmeettinen. Voit halutessasi laskea osioiden koot itse täyden varmuuden saamiseksi.

Mikäli käytössä on AmigaOS 3.5 tai 3.9, käyttöjärjestelmän mukana tuleva HDToolBox vaatii ReAction-käyttöliittymäkirjastot. Kirjastot tulevat asennuksessa mukana, mutta jos aiot tehdä käynnistyslevykkeen HDToolBoxia varten,

joudut kopioimaan enemmän tiedostoja HDToolBoxin kaveriksi. Viimeisissä HDToolBox-versioissa on korjattu suurten lukujen esitysongelma, joten suosittelen toki niiden käyttöä, mikäli aiot ajaa viimeisiä käyttöjärjestelmäversioita.

Tämä artikkeli keskittyy kuitenkin klassiseen HDToolBoxiin. Uudemmissa ReAction-versioissa toiminnot ovat hieman eri paikoissa mutta käytetyt termit ovat pääpiirteittäin samoja.

Vanhempien HDToolBox-versioiden kanssa on otettava huomioon se seikka, että ne eivät tue Removable-tilassa tunnistuvia massamuisteja. Aminetistä löytyy paikkatiedostot tätä varten paketista HDToolBoxPatch.lha. Tarvitset lisäksi paketin gpatch.lha. Paikkatiedostoja ei löydy aivan ensimmäisiin HDToolBox 1.0 ja 1.1 -versioihin, jotka tulivat A590- ja A2091-ohjainten asennuskorpuilla, mutta tämä ei liene suuri menetys. Paikkauspaketista löytyvässä .guide-hypertekstissä on mukana myös ohjeet manuaalista paikkausta varten, mikäli haluat muuttaa tavut heksaeditorilla ilman gpatchia.

HDInstTools

Toinen suosittu työkalu on Oliver Kastlin kirjoittama HDInst Tools. Se on helppokäyttöinen, osaa näyttää suuretkin luvut näitisti ja hallitsee myös kaikki RDB:n ominaisuudet. Tämäkin löytyy Aminetistä paketista hdinst.lha.

HDInstTools on muuten hyvä työkalu, mutta siihen on jäänyt yksi vaarallinen bugi. HDInstTools laskee levyn geometrian aina omalla algoritmillaan, ja mikäli olet alustanut levyn jollain toisella työkalulla, RDB:ssä oleva geometria ei välttämättä vastaakaan HDInstToolsin itse laskemaa geometriaa. Mikäli geometriat eivät täsmää, kaikki toisella työkalulla luomasi osiot ovat HDInstToolsin mielestä pielessä, ja se tuhoaa ne, kun tallennat muutokset.

Tämän johdosta HDInstToolsia tulee käyttää vain tyhjiä levyille tai sellaisille levyille, joiden tiedät olevan samalla ohjelmalla alustettuja aiemmin. Tämän vuoksi en myöskään voi ihan varauksetta suositella sen käyttöä, vaikka se muuten hyvä työkalu onkin.

Temppu ja miten se tehdään

Varsinainen osiointiprosessi voidaan suorittaa joko oikealla Amigalla tai emulaattorilla. Mikäli sinulla on mahdollisuus kirjoittaa Amiga-korppuja, on prosessi mielenkiintoinen tehdä aidolla koneella. Mikäli tätä mahdollisuutta ei ole, emulaation käyttäminen jää helpommaksi ja luultavasti myös ainoaksi vaihtoehdoksi.

Ensiksi tulee haalia tarvittavat työkalut kasaan:

- HDToolBox 40.4 (paikkaa tarvittaessa)
- pfs3_aio-handler
- Version-käsky
- Format-käsky
- Info-käsky.

Mikäli mennään yli 7,8 gigatavun rajan, tarvitaan lisäksi koneesi kanssa yhteensopiva paikattu scsi.device sekä LoadModule-käsky.

Valmistelutoimet Amigalla

Asenna kiintolevy tai CF-kortti paikalleen. Boottaa Workbench-levykkeeltä ja varaa tyhjä levyke valmiiksi. Huomaathan, että Workbenchin major-versio ei saa olla isompi kuin Kickstart-versiosi, sillä muuten osa käskyistä kieltäytyy toimimasta. Esimerkiksi Workbench 2.x ei toimi Kickstartin 1.x-versiossa.

Formatoi tyhjä levyke ensimmäisessä asemassa ja tee siitä boottaava Install-komennolla.

```
1.> copy c:install ram:
1.> sys:system/format drive df0: name BOOT noicons
```

Vaihda tyhjä korppu asemaan ja paina Return, kun Format niin kehottaa.

```
1.> ram:install drive df0:
```

Install-käskey ei kysele mitään vaan luo boottisektorin heti, joten käskey kopioitiin c:stä ram-levylle. Luonnollisesti asemien valinta on vapaaehtoinen. Käytä niitä asematunnuksia, jotka vastaavat omaa järjestelmääsi. Jos sinulla on lisälevyasema, työlevyke on helpointa pitää asemassa DF1.

Luo uudelle levyllä hakemisto devs ja kopioi Workbench-tai Install-levyltä devs/system-configuration uuden korpun devs-hakemistoon.

```
1.> makedir boot:devs
1.> copy devs:system-configuration boot:devs
```

Kopioi tarvittavat työkalut ja handler paikalleen.

```
1.> copy hdttools/hdtoolbox boot:
```

(Löytyy Install-korpulta HDTools-hakemistosta.)

```
1.> copy pfs3_aio-handler boot:
```

(Tämän olet haalinut itse jostain, esimerkiksi ClassicWB:n System.hdf:stä.)

```
1.> copy scsi.device boot:
```

(Tämäkin on siellä, mihin sen itse laitoit.)

```
1.> copy loadmodule boot:
```

(Tämäkin on sinun jäljiltäsi.)

```
1.> copy c:version boot:
```

(Löytyy Install- ja Workbench-korppujen c-hakemistosta.)

```
1.> copy c:info boot:
```

(Löytyy Workbench-korpun c-hakemistosta.)

```
1.> copy system/format boot:
```

(Löytyy Install- ja Workbench-korppujen system-hakemistosta.)

Huomaa: Edellä mainitut copy-käskyt luultavasti eivät toimi sellaisenaan, vaan sinun pitää kertoa copy-käskylle lähdetiedoston koko polku, vaikkapa df1:c/info, jos Workbench-levy on ensimmäisessä lisälevyasemassa, tai sys:c/info, mikäli käynnistit koneen Workbench-levykkeellä.

Käynnistä kone uudella levykkeellä ja siirry seuraavan kohdan yli yhteisiin ohjeisiin.

Valmistelutoimet WinUAE:lla

Liitä kiintolevy tai CF-kortti PC:hen haluamallasi tavalla.

Kerää kaikki edellä mainitut työkalut samaan hakemistoon Windowsin puolella. Tämä kannattaa tehdä vaikkapa ClassicWB:llä, niin saat kaikki kerralla kuntoon. Huomioi kuitenkin, että mikäli kohdekone on varustettu Kickstart 2.05:llä, sinun pitää kopioida käyttöjärjestelmästä tulevat tiedostot Workbench 2.x -levykkeiltä.

Klikkaa WinUAE-ikonia hiiren kakkospainikkeella ja

käynnistä se järjestelmänvalvojana.

Valitse quickstartista konettasi vastaava pohjakonfiguraatio. Scsi.device on konekohtainen, joten valinnan täytyy täsmätä. MKLIDE- tai Kipper IDE -käyttäjät valitsevat A600:n.

Mene tämän jälkeen CD & Harddrives -osioon ja liitä työkaluhakemistosi kiintolevyksi painamalla Add Directory or Archive. Laita device- ja volume-nimiksi BOOT sekä poluksi täysi polku kyseiseen hakemistoon Windowsin puolella. Liitä samalla Amigaan tuleva massamuisti kiintolevyksi Add HardDrive -nappia painaen. Valitse alusvetolistasta oikea massamuisti ja valitse HD Controlleriksi IDE 0. Tämä on tärkeää, jotta scsi.device:n päivitys voidaan testata jo emulaatiossa.

Voit tallentaa nyt konfiguraation menemällä Configurations-osioon ja valitsemalla sille nimen. Tämä kannattaa tehdä, jotta ei tarvitse aloittaa alusta, jos homma keskeytyy jostain syystä.

Boottaa nyt emulaattorin sisällä oleva Amiga. Sen pitäisi käynnistyä BOOT-hakemistostasi.

Yhteiset ohjeet

Kun Amiga on käynnistetty tekemältäsi boottikorpulta tai emuloidusta BOOT-hakemistosta, voit jatkaa tästä.

Voit ensin tarkastaa scsi.device:n version:

```
1.> version scsi.device
```

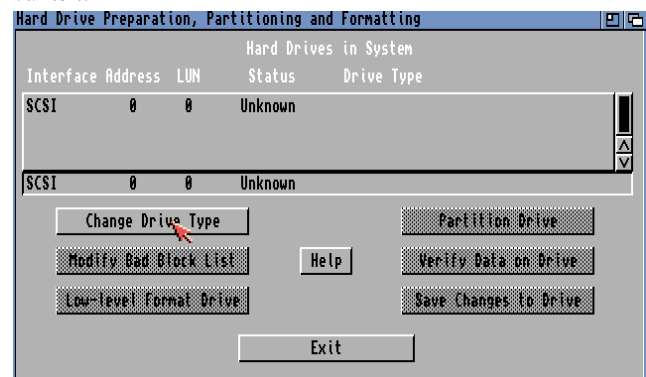
Paina vanha versio mieleesi tai kirjoita se muistiin. Sitten voit ladata uuden scsi.device:n:

```
1.> loadmodule scsi.device
```

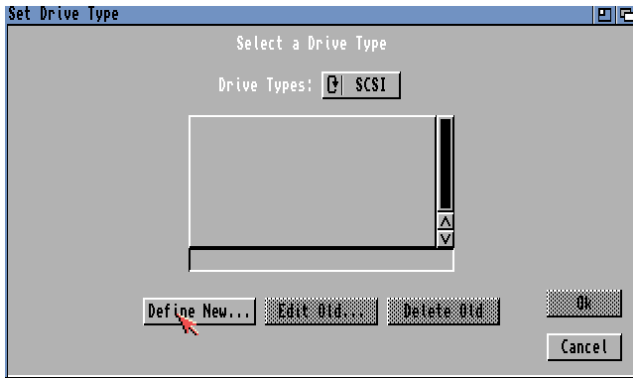
Amiga käynnistyy uudestaan ja palauttaa sinut komentokehotteeseen. Kannattaa tarkastaa scsi.device:n versio uudestaan ennen kuin jatkat.

```
1.> version scsi.device
```

Nyt pitäisi näkyä 43.24 tai uudempi, riippuen siitä minkä valitsit.

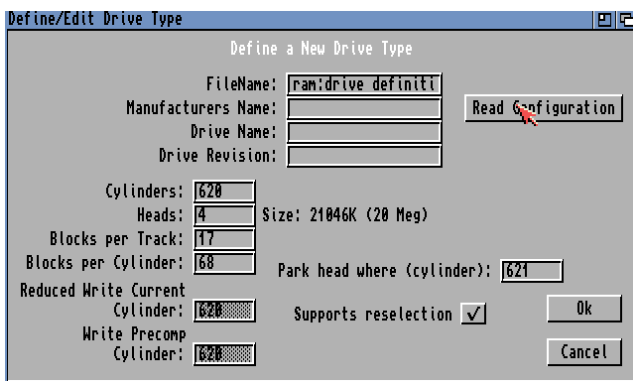
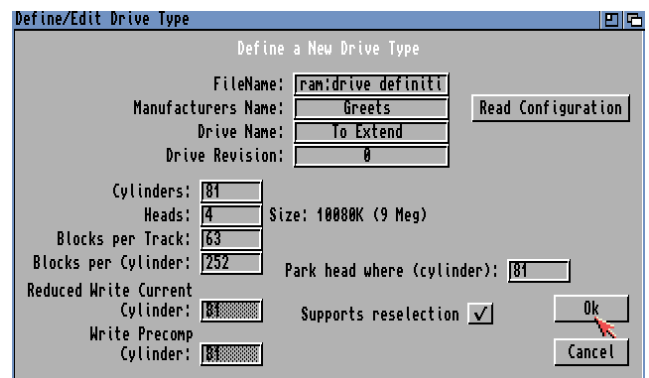


Seuraavaksi voit käynnistää HDToolBoxin ja aloittaa varsinaisen osioinnin. HDToolBoxissa, kuten monissa muisakin vanhemmissa Amiga-ohjelmissa, täytyy muistaa, että tekstikentän muokkaukset pitää aina kuitata Return-näppäimen painalluksella. Return ei sulje nykyistä näkymää kuten Windowsissa vaan ainoastaan vahvistaa aktiivisen tekstikentän muutokset. Jos et muista painaa Returnia, saat- taan joihinkin kenttiin jäädä oletusarvo ja joudut vianhaku- puuhiin.



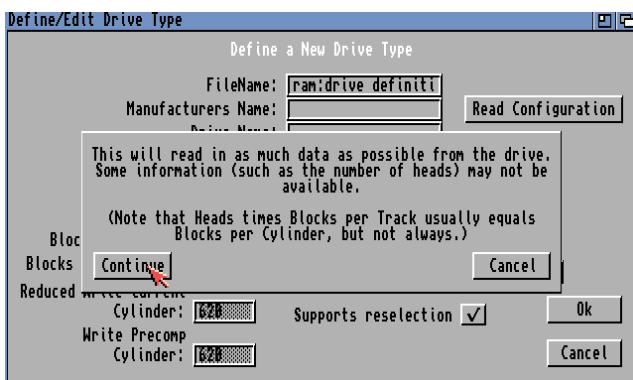
Aivan ensimmäiseksi pitää lukea levyn geometria. Levyn pitäisi näkyä HDToolBoxin listassa Unknown-statuksella. Valitse tämä tuntematon levy ja sen jälkeen paina nappulaa Change Drive Type. Pääset Set Drive Type -näkömään, jossa oikeastaan ainoa vaihtoehtosi on Define New.

kanssa, voit jatkaa hyvillä mielin. Saatat tarvita prosessin aikana geometrialukuja myöhemminkin, joten nuo kolme muuttujaa kannattaa kirjoittaa muistiin ennen kuin jatkat.

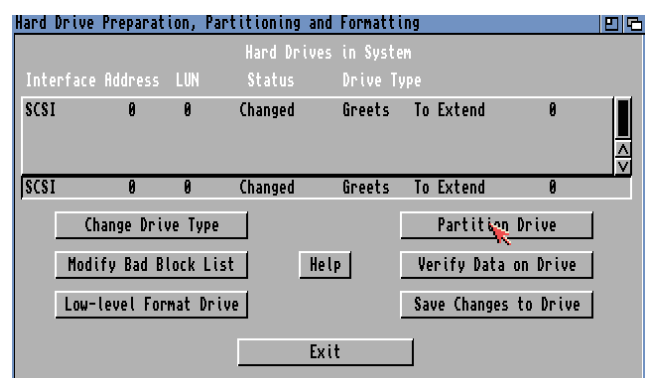


Seuraavassa näkymässä FileName-laatikon sisältöön täytyy tehdä muutoksia. HDToolBox tallentaa aiemmin tunnistetut levyt tiedostoon drive definitions. Klikkaa tekstin alkuun ja vaihda sisällöksi ram:drive definitions. Näin toimimalla tiedosto luodaan haihtuvalle RAM-levylle. Voit toki myös luoda sen käynnistyslevyllesi, mikäli se ei ole kirjoitussuojattu, ja tällöin voit jättää arvon muuttamatta. Levyjen määritelmillä ei kuitenkaan tee nykypäivänä mitään, kun kaikki varteenotettavat kiintolevyt osaavat kertoa itse tietonsa.

Ok-nappi palauttaa sinut edelliseen näkömään, ja listassa pitäisi olla valmiiksi valittuna tämä vasta tunnistamasi levy. Paina jälleen Ok ja pääset alkuruutuun. Koska et kopioinut FastfileSystemiä työlevyllesi, HDToolBox valittaa tässä yhteydessä, että FastfileSystemiä ei löydy. Voit ohittaa tämän Continue-painikkeella, sillä emme tarvitse FastfileSystemiä.

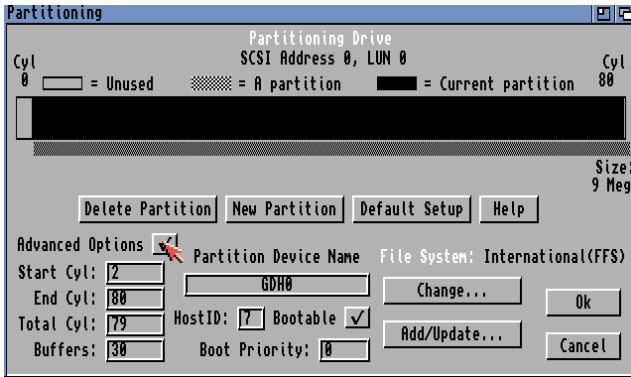


Kun FileName-kenttä on muutettu (ja Returnilla kuitattu), valitse "Read configuration" ja kuittaa ikkunan keskelle ilmestynvä valintaikkuna. Nyt sinulla on levyn geometriarvot luettuna ja voit valita Ok. Jos Size-kohdassa lukee jostain muuta kuin mitä oletit, voit laskea itse koon kaavalla cylinders × heads × blockspertrack × 512. Mikäli oma las- kutoimituksesi täsmää kiintolevyn kanssa lukevan koon



Levy paloihin

Seuraavaksi mennään Partition Drive -näkyämään suorittamaan varsinainen osiointi. Aivan ensimmäiseksi laita valintamerkki Advanced Options -vaihtoehtoon, sillä meidän täytyy myöhemmin käydä muuttamassa näitä lisävalintoja.



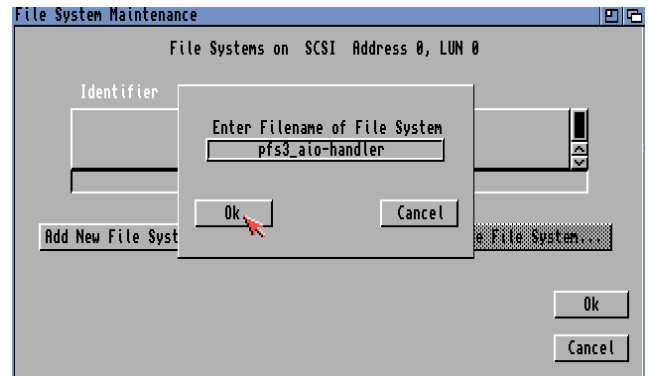
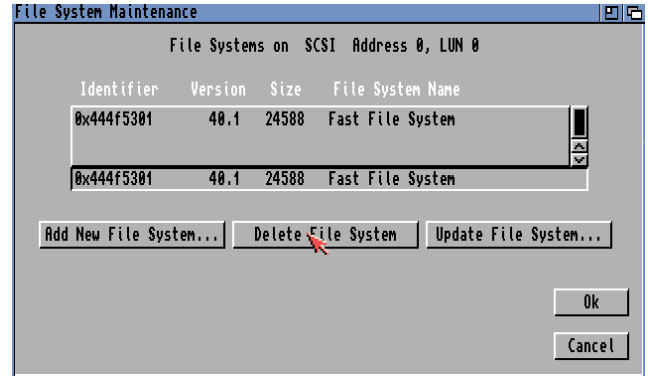
Riippuen levyn koosta HDToolBox laittaa sinne ehdotukseksi yhden tai useamman osion. Voit nyt venyttää osioita haluamallasi tavalla ja lisätä tai poistaa niitä tarpeesi mukaan.

Jostain syystä HDToolBox ottaa levyn valmistajan ensimmäisen kirjaimen Device Nimen alkuun, esimerkiksi Seagaten levyille tehdään SDH0, SDH1 jne. Standardi on kuitenkin DH0, DH1 jne., joten itse olen tavannut poistaa ylimääräisen alkukirjaimen osioista. Muista painaa Return muutosten jälkeen.

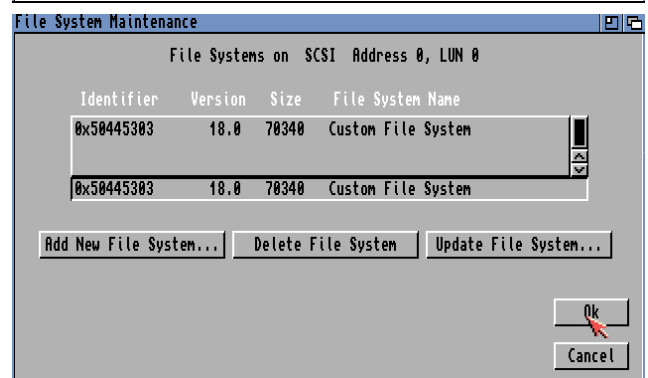
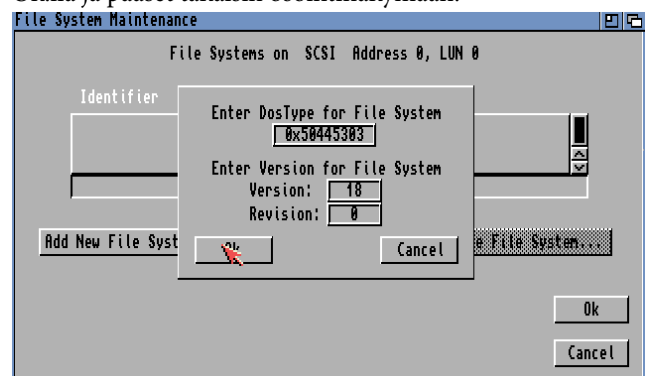
Jälleen kerran suurilla levyillä saattaa tulla negatiivisia ja liian pieniä lukemia, joten voit halutessasi laskea varsinaiset koot itse kaavalla $\text{total cyl} \times \text{heads} \times \text{blocksptrack} \times 512$. Heads ja blocksptrack kirjoitettiin talteen, kun koko levyn geometria luettiin.



Seuraavaksi asennetaan tiedostojärjestelmä RDB:hen, jotta se on käynnistyksen yhteydessä käytettävissä. Valitse osiointinäkyössä Add/Update. Pääset valitsemaan tiedostojärjestelmää. Mikäli boottikorpullasi olikin FFS, se on nyt automaattisesti asennettuna. Tässä tapauksessa valitse ja poista se ja klikkaa sitten Add New File System.

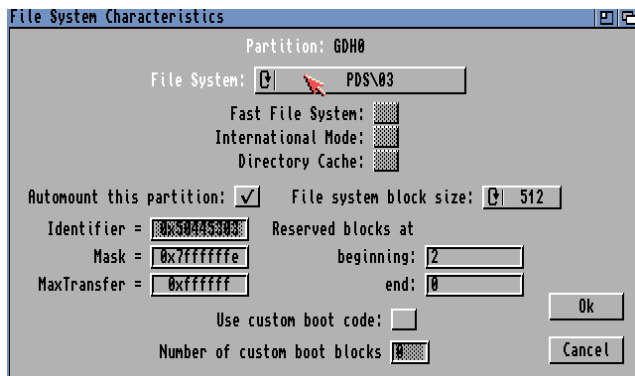


Tyhjennä tekstikenttä ja kirjoita siihen pfs3_aino-handler. Valitse sitten Ok. HDToolBox lukee käsittelijän levytä muistiin ja antaa mahdollisuuden asettaa dostyypin sekä versio- ja revisiotiedot. Näistä dostyypin on tärkeä. Tiedostojärjestelmiä voi olla muistissa yhtä aikaa useita, ja dostyypin avulla kiintolevyajuri tietää, mitä järjestelmää kuhunkin osioon tulee käyttää. Koska tavoitteenamme on käyttää DirectSCSI:tä, dostyypeksi tulee PDS\03, eli kenttään kirjoitetaan 0x50445303. Jos käyttäisimme TD64:ää, dostyypeksi tulisi PFS\03 eli 0x50465303. Muista jälleen Returnin painallus ja valitse Ok. Kuittaa myös Add/Update-ikkuna pois Ok:lla ja pääset takaisin osiointinäkyymään.

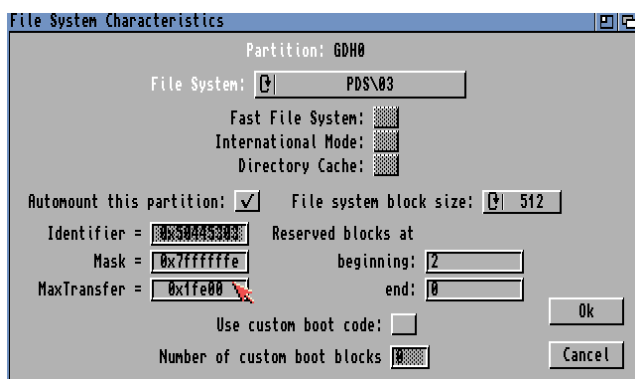
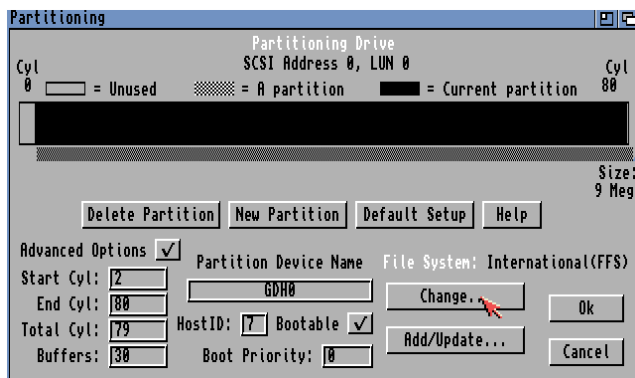


Max siirtelee

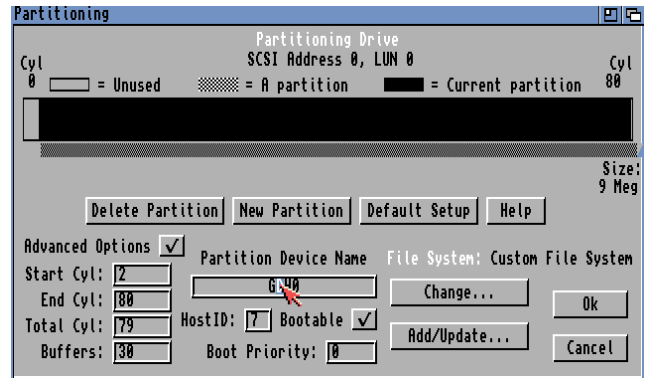
Seuraavaksi onkin aika asettaa osiolle tiedostojärjestelmä ja muuttaa MaxTransfer-koko sopivaksi. Tee tämä jokaiselle osiolle.



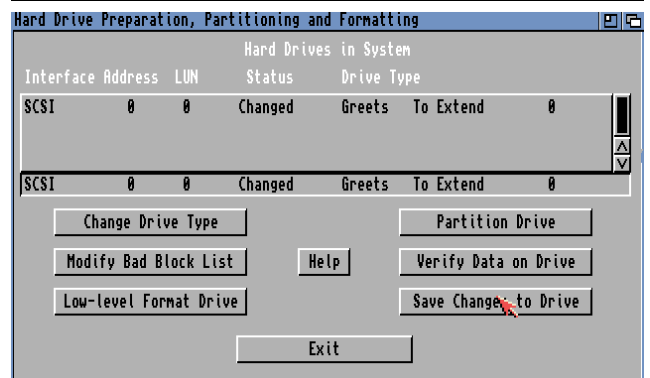
Valitse osio ruudun yläreunassa olevasta palkista ja paina Change-nappulaa. Seuraavan ruudun yläreunassa on kiertovalinta nimeltään File System. Napsuttele valintaa, kunnes ruudussa lukee aiemmasta riippuen joko PDS\03 tai PFS\03. Näin tiedostojärjestelmä on valittu.



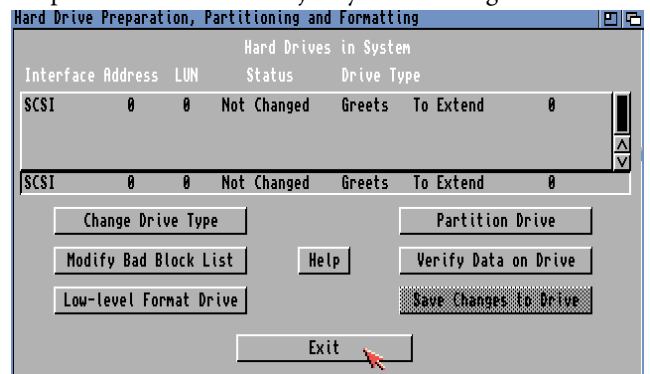
Seuraava hyvin tärkeä asetus on MaxTransfer. IDE-protokolla on aikojen saatossa muuttunut, mutta scsi.device on jämähtänyt vuoteen 1992, joten se ei ole uusien massamuistien kanssa täysin yhteensopiva. Kiintolevyn tapa kuitata suuret tiedonsiirrot on muuttunut aikojen saatossa, eikä scsi.device ymmärrä uudemmallista vastausta.



Kiertoreittinä voimme MaxTransfer-asetuksella rajata siirrettävät blokit riittävän pieniksi, että siirrettävä tieto mahtuu yhteen IDE-käskyyn eikä bugiin törmätä. Tyhjenä tekstikenttä ja vaihda siihen 0x1fe00 (muista Return). Muuta ei tarvitse osiolle tehdä. Voit kuitata tämän näkymän Ok:lla ja siirtyä tekemään samat temput seuraavalle osiolle. SCSI-ohjaimen käyttäjien ei tarvitse olla niin huolissaan MaxTransferista, sillä siellä ei vastaavaa muutosta ole tapahtunut.



Kun nämä asetukset on muutettu kaikkiin osioihin, voit palata alkuunäkymään ja valita Save Changes to Drive. Sitten voitkin poistua HDToolBoxista ja käynnistää Amigan uudestaan.



Aika alustaa

Kun kone on käynnistynyt uudestaan, voit listata kiintolevyajurin löytämät loogiset laitteet Info-käskyllä:

```
1.> info
```

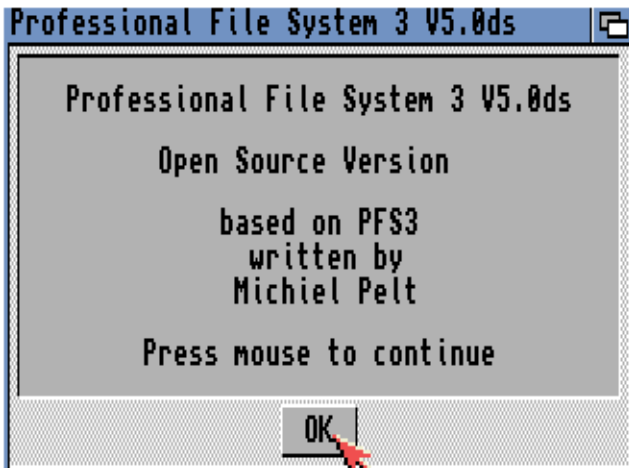
Kaikkien aiemmin luomiesi osioiden tulisi näkyä listassa. Jos niin ei ole, käy HDToolBoxilla tarkistamassa, että dos-type on tiedostojärjestelmässä ja kussakin osiossa oikein ja muuttamiesi tekstikenttien muutokset ovat menneet perille. Jos jokin muutos ei ensimmäisellä kerralla ollut tarttunut, tee muutos uudestaan ja muista painaa Returnia joka kerta. Tallenna sitten muutokset levyille ja käynnistä Amiga uudestaan.

Kun osiot viimein näkyvät, alusta ne:

```
1.> format drive dh0: name HardDrive0 noicons quick
```

Format-käskyn drive- ja name-vivut ovat pakollisia. Komentorivillä dh0:n tilalle vaihdat alustettavan osion nimen ja HardDrive0:n tilalle haluamasi pitkän nimen osiolle. Noicons-valinta on vapaaehtoinen, ja se määrittelee, tuleeko levyille roskapönttöä. Itse jätän roskiksen yleensä pois, sillä Workbench-asennus luo sen kuitenkin myöhemmin.

Quick-valinta on hyvin tärkeä. Useimmat Amigan Format-käskyistä eivät ymmärrä yli neljän gigatavun osioita. Quick Format sen sijaan on aina turvallinen, sillä se pyytää tiedostojärjestelmää suorittamaan alustuksen. Kaiken lisäksi se säästää paljon aikaa myös alle 4 Gt:n rajan jäävillä osioilla. Mikäli sinulla on SSD-asema tai flash-kortti, et varsinkaan halua, että Format-käsky kirjoittaa levyn täyteen nollaa, sillä se kuluttaa flash-soluja ja merkitsee niitä varauksiksi levyaseman sisäisen ohjaimen mielestä.



Kun alustus on valmis, pitäisi ruutuun tulla pfs3:n tietoikkuna, jossa kerrotaan versionumero ja tekijöiden nimet. Kuittaa tämä ja siirry alustamaan seuraavia osiota.

Työpöytä kohti

Kun kaikkien uusien osioiden alustus on tehty, voit siirtyä käyttöjärjestelmän asentamiseen. Jos haluat puhtaan Workbenchin, boottaa Install-levyltä, avaa sen sisältä Install-hakemisto ja aloita asennus haluamallasi kielellä.

Mikäli sinulla on jokin olemassa oleva asennus emulaattorin sisällä, liitä uusi kiintolevyysi tämän asennuksen konfiguraatioon, käynnistä emulaattori ja kopioi tiedostot seuraavalla käskyllä:

```
1.> copy lähdeosio: kohdeosio: all clone
```

All-vipu kopioi tiedostot ja hakemistot rekursiivisesti, ja clone säilyttää tiedostojen kommentit ja suojausbitit ennallaan. Info-käskyllä näet osioiden nimet Unit-sarakkeen alla.

Suosittelen kopioinnin yhteydessä käyttämään osioiden lyhyitä nimiä, mikäli Name-kentässä näkyvät pitkät nimet sattuvat olemaan samat kahden eri osion kesken. Kiintolevyajurit huolehtivat siitä, että osioiden nimet ovat yksilöllisiä lisäämällä väliaikaisesti kirjaimia tai numeroita, kunnes kaikki osiot ovat eri nimiä keskenään. Tämä ei tee muutoksia RDB:hen, vaan toiminto on muistinvarainen.

Muista kopioida LoadModule kohdeosion c-hakemistoon ja paikattu scsi.device kohdeosion devs-hakemistoon.

Muokkaa kohdeosion s/startup-sequence-tiedostoa:

```
1.> ed kohdeosio:s/startup-sequence
```

Lisää ensimmäiseksi riviksi seuraava:

```
c:loadmodule devs:scsi.device
```

Tallenna muutokset: Esc x Return.

Odota muutama sekunti, että pfs3 tekee atomisen päivityksen (kiintolevyvalo vilahtaa pienen tauon jälkeen), ja sitten homma on valmis. Jos teit operaation emulaattorilla, siirrä levy oikeaan Amigaan ja kokeile. Jos teit sen oikealla Amigalla, katkaise virrat 30 sekunniksi ja yritä kylmäkäynnistystä uudelta kiintolevyltäsi.

Mikäli ongelmia ilmenee, kannattaa etsiä vaikkapa archive.org-sivustolta Amigan käsikirjat ja lukea myös Amignetistä löytyvän alkuperäisen pfs3-paketin mukana tuleva dokumentaatio. Myös internetin eri foorumit sisältävät hyvin paljon tietoa ja esimerkkejä tämän asian tiimoilta.

Onnea matkaan, ja muista edetä rauhallisesti. Prosessi ei ole kovin vaikea, mutta hieman työläs se kieltämättä on. 🐼

Linkit tarvittaviin tiedostoihin

- <http://aminet.net/package/driver/media/SCSI4345p>
Scsi.device 43.35 -paikat
- <http://os.amigaworld.de/index.php?lang=en&page=37>
Scsi.device 43.24 -paikat
- <http://aminet.net/package/util/sys/patchstrip>
Scsi.device 43.24:n aikarajan poistopaikka.
- <http://aminet.net/package/disk/misc/HDToolBoxPatch>
HDToolBox type 7 -paikat.
- <http://aminet.net/package/util/misc/gpatch>
Paikkatiedostojen asennusohjelma.
- <http://aminet.net/package/disk/misc/pfs3aio>
PFS3aio.
- <http://aminet.net/package/util/boot/LoadModule>
LoadModule-käsky.
- <http://www.amigaforever.com/shop/>
1.x–3.1 käyttöjärjestelmien levykuvat (Image Pack) online-kaupasta.
- http://thomas-rapp.homepage.t-online.de/4gb_faq.html
Thomas Rappin hyvä UKK-sivu aiheesta englanniksi.
- <http://thomas-rapp.homepage.t-online.de/filesyslimits.html>
Thomas Rappin tekemä suhteellisen ajantasainen taulukko eri tiedostojärjestelmien ja kiintolevyajureiden kokorajoista.



Everdrive

Loputtomasti pelattavaa klassikkokonsolille

Olet vihdoin saanut hankittua takaisin lapsuutesi suosikkikonsolin ja kaikki rakastamasi pelit. Pian kuitenkin huomaat, että menneessä ajassa oli myös huonot puolensa: isokokoisia pelejä on hankala kuljettaa ja niiden toimivuus on oikukasta. Mitenköhän voisi saada parhaat puolet sekä uudesta että vanhasta?

Teksti ja kuvat: Mikko Heinonen

Ohjelmoitavat FPGA-piirit ovat mullistaneet lyhyessä ajassa harrastelijoiden elektroniikkaprojektit. Ennen jokainen piiri piti valmistaa erikseen, mutta nykyisin logiikkaa voidaan rakentaa FPGA:n sisään erityisellä kuvauskielellä. Samalle piirille mahtuu useita eri logiikkayksiköitä. Tämä on laskenut huomattavasti erilaisten laitteiden suunnittelu- ja valmistuskustannuksia ja tehnyt pientenkin sarjojen tuottamisesta kannattavaa.

Kehityksestä ovat hyötynneet monin tavoin myös vanhojen laitteiden harrastajat. Viime vuosina on käynnistynyt useita projekteja, joissa vanhoihin kotimikroihiin ja pelikonsoleihin on sovitettu nykyaikaisia tallennusvälineitä. Commodore 64:n omistajat ovat saaneet muun muassa 1541 Ultimaten, joka matkii pikkutarkasti levyasemaa, eivätkä muutkaan laitteet ole jääneet osattomaksi. Amiga 600:lle tarkoitettu ja Skrollissakin esitelty Vampire-turbokortti perustuu myös kokonaan FPGA-tekniikkaan.

läisyasema

Pelikonsoleille on jo pitkään ollut saatavana erilaisia flash-muistiin perustuvia moduuleita, joille voidaan kirjoittaa pelejä tai muita ohjelmia ja sitten ajaa niitä alkuperäisessä laitteessa. Niissä on kuitenkin ilmeiset rajoituksensa. Moduulin flash-muistiin mahtuu vain rajallinen määrä sisältöä kerrallaan, ja se on aina ohjelmoitava uudelleen jonkin isäntälaitteen avulla.

Everdrive on ukrainalaisen Igor

”Krikzz” Golubovskin projekti, joka sai alkunsa Sega Megadriven muistikortinlukijasta. Se sisältää flash-muistin lisäksi laitteiston ja ohjelmiston, joka lukee halutun tiedoston SD-kortilta, kirjoittaa sen flashiin ja käynnistää sitten pelin varsinaisella konsolilla.

Koska Everdrive on älykäs lisälaitte, se mahdollistaa pelkän flash-muistin käytön lisäksi myös monia muita temppuja. Monissa malleissa pelitilanne on mahdollista tallentaa ja ladata kesken pelin aivan kuten emulaattoria käytettäessä. Samoin voidaan käyttää erilaisten cheat-moduulien koodeja ja muokata ROM-tiedostoa muillakin tavoin. Everdrive voi toimia myös esimerkiksi Megadriven CD-aseman ohjaimena, ja se osaa ladata 32X-laajennuksen pelejä siinä kuin tavallisiakin Megadrive-pelejä.

Muutkin harrastajat ovat tuottaneet vastaavia moduuleita eri laitteille aina Atari 2600:sta alkaen. Everdriven erottaa useimmista muista vastaavista projekteista se, että Krikzz on ilmeisen tosissaan tuotteidensa suhteen ja palvelu on ammattimaista. Toimitukset ovat nopeita, uusia laiteohjelmiston versioita julkaistaan säännöllisesti eikä vanhojakaan tuotteita hylätä. Päivittäminen on myös helppoa, sillä uusi ohjelmisto vain tallennetaan SD-kortille, josta Everdrive noukkii sen käynnistuksen aluksi.

Piratismia vai harrastamista?

Periaatteessa Everdriveä voi hyödyntää pelkästään täysin laillisiin tarkoituksiin. Sen mukana ei toimiteta mitään

tekijänoikeudellisesti arveluttavaa, eikä laitteen käyttäminen vaadi mitään muutoksia konsoliin. Jos haluaa vaikkapa katsella eri konsoleille tehtyjä demoja tai homebrew-pelejä, tai peräti kehittää sellaisia itse, Everdrive on loistava laite ohjelmien testaamiseen oikealla laitteistolla.

On kuitenkin turhaa teeskennellä, etteikö Everdriveä useimmiten hankittaisi kaupallisten pelien pelaamista varten. Jokainen voi itse päättää, miten suhtautuu moraalisisella tasolla yli 20-vuotiaiden pelien ROM-dumpien pelaamiseen sen sijaan, että kantaisi mukanaan valtavaa kasaa alkuperäisiä pelimoduuleita. Lain silmissähän tämä on luonnollisesti tekijänoikeusrikkomus, etenkin jos kopiot valmistaa itse.

Virallisten pelien lisäksi Everdrive lataa tietenkin myös erilaisia ROM-hackeja, joita pelaajayhteisössä on tuotettu. Näihin kuuluvat muun muassa japanilaisten pelien fanikäännökset, joita ei oikealla moduulilla ole koskaan julkaistukaan.

Testissä neljä Everdriveä

Megadriven jälkeen Everdrive on valloittanut muitakin konsoleita. Tällä hetkellä se on saatavana myös NESille, Super Nintendo, Nintendo 64:lle, Game Boylle, Game Boy Advancelle, Sega Master Systemille, Game Gearille ja PC Engineille. Ominaisuudet vaihtelevat hieman konsolista riippuen, mutta peruseräaate säilyy samana.

Kokeilin neljää eri Everdriveä: Megadriven Everdrive-MD v3:a, SNESin Super Everdriveä, PC Enginen Turbo Everdrive v2:ta sekä Everdrive N8:n Famicom-versiota. Kaikki löytyvät Krikzzin omasta kaupasta osoitteessa shop.krikzz.com. Koti-Suomesta käsin voi tulla vaikea käsittää, että samassa maassa käydään toisaalla repivää sotaa ja toisaalla tehtaillaan pelikoneiden flash-moduuleita, mutta Ukraina on laaja valtio, jossa alueelliset erot ovat suuria.



Super Everdrive

Super Nintendo on hieman erilainen eläin kuin Sega-merkkinen kilpailijansa. Siinä missä Segalla moduulit sisälsivät vain harvoin mitään lisäominaisuuksia, SNESillä tällaisia pelejä on useita kymmeniä. Kokeilemani Super Everdrive ei sisältänyt vakiona DSP:tä, mikä tarkoittaa käytännössä, että sitä tukevat pelit, kuten vaikkapa Mario Kart ja Pilotwings, eivät toimi.

Tämä laskee hieman laitteen hyödyllisyyttä, jos sitä on tarkoitus käyttää pelaamiseen. Piirilevyllä on kuitenkin paikka jälkiasennettavalle DSP:lle. Se on mahdollista hankkia 20 dollarin lisähintaan tai kähveltä sopivasta pelimoduulista. Jälkimmäinen vaihtoehto vaatii hieman juottamista.

Viimeksi Japanissa käydessäni löysin erään pelikaupan "romulaarista" hieman kärsineen Pilotwings-pelin, joka maksoi alle kaksi euroa. Minulle tuli vahva tunne, että tämä kannattaa poimia mukaan. Intuitio osoittautui oikeaksi, sillä Pilotwings sisältää DSP-piirin, eikä rupista moduulia jää moni kaipaamaan.

Avasin moduulin Nintendo-erikoistyökälulla ja irrotin DSP:n juotokset käyttäen lämmittävää tinaimuria. Lopuksi puhalsin piirin jalkoihin kuumailmaa ja vedin sen PLCC-ulosvetäjällä irti. Ulosvetäjän käyttö auttaa osaltaan siinä, että piirin jalat eivät mene vinoon.

Itse juottaminen on varsin helppoa, kunhan huomaa katsoa, miten päin piiri asennetaan (kääntöpuolelle muihin osiin nähden). Tämän operaation jälkeen ainakin Super Mario Kart ja Pilotwings toimivat SD-kortilla ladattuina komeasti, mutta puutteita peliyhteensopivuuteen jää edelleen. Muun muassa Super FX -piiriä käyttävät pelit, kuten Starfox ja Super Mario World 2, eivät toimi.

Erikoispiireistä johtuvaa säätöä lukuun ottamatta Super Everdrive toimii hyvin ja varsin samanlaisesti kuin Everdrive-MD. Se on myös hinnaltaan samaa luokkaa, noin 80 dollaria. Lähes tuplahintaan on saatavana SD2SNES, joka sisältää myös DSP-piirejä matkivan FPGA:n. Se tukee jo nyt suurinta osaa eri erikoispiireistä ja myös Super FX -tuki on tekeillä.



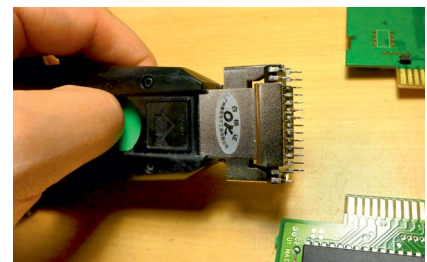
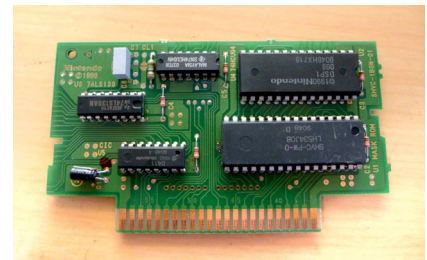
Everdrive-MD v3

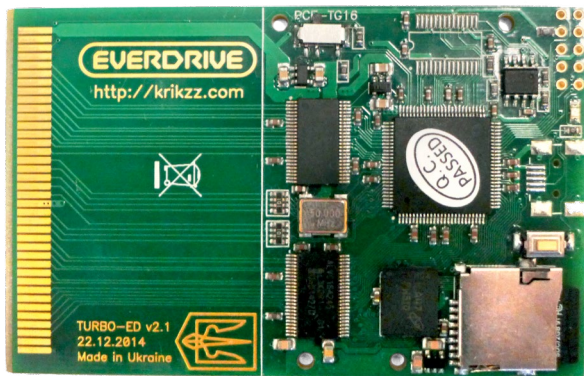
Krikzz aloitti kehitystyönsä Sega Megadriven kortinlukijoista. Everdrive-MD onkin selkeästi toimiva laite ja sisältää kaikki tarvittavat ominaisuudet. Virran kytketyessä se lataa näkyviin luettelon SD-kortilla olevista ROM-tiedostoista, joista voi poimia haluamansa ja siirtää sen moduulin flash-muistiin. Käyttöliittymä toimii Megadriven ohjaimella.

Flash-muistin pyyhkiminen ja kirjoittaminen kestää hetken, hieman ROM-tiedostosta riippuen. Kätevä ominaisuus on se, että valmiiksi muistiin ladatun ohjelman voi käynnistää Start-painikkeella, vaikka virran olisi välillä katkaissut. Everdrive-MD osaa ohittaa pelien aluelukitukset (paitsi 32X:n peleissä), ja sille voi syöttää myös cheat-moduulin koodeja. Koska Megadrive on myös Sega Master System -yhteensopiva, onnistuu Master System -pelienkin pelaaminen samalla moduulilla. Muutamat lisäroutaa sisältävät pelit, kuten Virtua Racing, eivät toimi.

Pelien lataamisen lisäksi Everdrive-MD:tä voi käyttää muihinkin tarkoituksiin. Jos käytössä on Mega-CD-lisälaite, Everdrivellä voi muuttaa sen aluekoodauksen yksinkertaisesti lataamalla muistiin toisen alueen BIOSin. Tämä toimii jopa Pioneer LaserActivessa, joka yleensä hylkii kaikenlaisia aluevapausvirityksiä.

Saatavana on myös kehittyneempi versio Mega Everdrive. Se tukee muun muassa savestatea eli pelitilanteen tallentamista vedokseksi muistiin. Lisäksi pelien mainitaan latautuvan sillä flash-muistiin parissa sekunnissa, ilman odottelua. Mega Everdriven hinta on kuitenkin korkeampi: Everdrive-MD maksaa noin 80 dollaria, Mega Everdrive yli 140 dollaria.



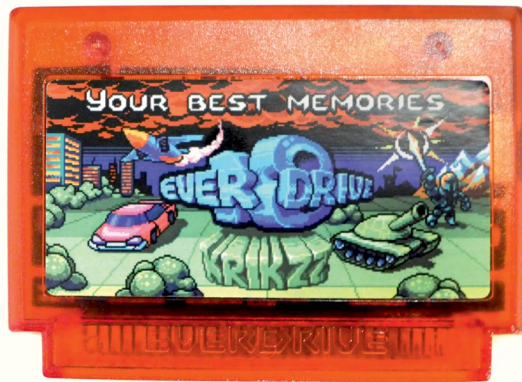


Turbo Everdrive v2

NEC PC Engine tunnettiin lännessä nimellä TurboGrafx, josta tämän laitteen nimi juontaa juurensa. Toisin kuin useimpiin muihin Everdriveihin, Turboon ei saa koteloaa. PC Enginen pelit toimitettiin pankkikortin kokoisilla HuCardeilla, ja laitteen piirilevy on jo yksinään yhtä paksu. Ratkaisu on kuitenkin vähemmän huolestuttava kuin voisi ajatella, ja piirikortti tuntuu riittävän tukevalta.

Laitetuki on erittäin hyvin kunnossa: kortti toimii niin perus-Enginellä, kehittyneemmällä SuperGrafxilla kuin LaserActiven NEC-lajennuksella. Läntisen TurboGrafxin moduuleissa on eri nastajärjestys, joten Everdrivestä pitää valita kytkimellä, käytetäänkö sitä japanilaisen vai amerikkalaisen/eurooppalaisen laitteen kanssa. CD-aseman ohjelmisto toimitettiin tässä konsolissa HuCardilla, joten Everdrive korvaa myös tämän "System Cardin" tarvittaessa.

Käytettävyydeltään Turbo vastaa muita malleja. Se ei sisällä juuri erikoisominaisuuksia, mutta piirilevyllä oleva reset-kytkin on erityisen hyödyllinen, sillä PC Enginessä ei ole omaa resetiä. Pelit latautuvat huomattavan nopeasti, ja ohjelmistoyhteensopivuus on käytännössä täydellinen. Hinta on tutut 80 taalaa.



Everdrive N8

Kasibittisen Nintendon Everdrive on varmasti yksi suosituimmista malleista, ja ehkä sen vuoksi myös hinta-laatusuhteeltaan paras. Moduuli on saatavana sekä NESiin että Famicomiin ja hinta on molempiin 110 dollaria. Itse asiassa eri versioissa on erona vain liitin, sillä Famicom-peleissä on 60 nastaa, NES-peleissä 72. Ohjelmisto on molemmissa sama, ja sovitimella molemmat Everdrivet toimivat kummassa konsolissa tahansa. N8:n käyttöliittymä on erittäin sulava, ja ROMit käynnistyvät hetkessä.

Everdrive N8 on halvemmissa moduuleista ainoa, joka tukee savestea eli pelitilanteen tallennusta missä tahansa kohdassa. Kun toiminto on otettu käyttöön valikosta, tietyllä näppäinyhdistelmällä (esimerkiksi start+ylös) voi tallentaa tilanteen, toisella näppäilyllä (esimerkiksi start+alas) ladata sen. Tämä tapahtuu käytännössä välittömästi. Jotkut pelit sekoavat näin suorasta toiminnasta, mutta silloin, kun savestatet toimivat, voidaan puhua todella mullistavasta ominaisuudesta. Hankala tason loppuhirviö on huomattavasti helpompi peitota, kun tarjolla on loputtomasti yrityksiä. Tämän myötä Everdrive tuo loputkin emulaattoripelaamisen hyvät puolet myös oikealle raudalle. 🍌

TÖITÄ



ATK-OHJELMOITSIJOILLE!

Java / Node.js / Clojure / Python / C# / C++ / Scala / Ruby / TAMPERE / HELSINKI

GREAT PLACE TO WORK®

Suomen parhaat työpaikat 2015
Finland



GREAT PLACE TO WORK®

Best Workplaces 2015
Europe



VINCIT
www.vincit.fi



Nokian paluu

Stephen Elop puhelinyksiköineen siivottiin Microsoftilta ja Nokia palaa kännyköihin. Miten tässä näin kävi?

Janne Sirén

Microsoftin ostaman kännykkätoiminnan alasajo on alkanut, samalla kun Nokia ilmoitti paluustaan matkapuhelimiin. Tavallaan.

Hopeasurffari Elop

Aloitetaan alusta. Oliko Nokian toimitusjohtaja Stephen Elop troijanhevon, Microsoftin peiteagentti? Tuskin. Elopilla on toki uran mittainen maine yrityksiä pilkkovien ja kauppaavien galactusten esitaistelijana. Hän on lepton transitojohtaja, ja tappiollisen toiminnan myynti Microsoftille oli Nokian varaventiili, jota käytettiin, kun muutos epäonnistui.

Siihen se jääneekin. Microsoftin silloinen toimitusjohtaja Steve Ballmer varmasti kauppasi Microsoft-suhdetta Elopille, mutta tokkopa hän tätä hallitsi. Ballmer hädin tuskin hallitsi asiassa edes Microsoftia, jonka sisäiset valtaistelut tekivät yhtiön Nokia-ratkaisusta vastentahtoisen.

Nokia oli kunnianhimoiselle Elopille ura-askelma (perhe ei muuttanut mukana), joka onnistuessaan olisi avannut pikatien isompaan herrakerhoon. Epäonnistumisen myötä maine kärsi, avoliitto meni ja Microsoft-kokoluokan toimitusjohtajahaaveiden sijaan tulikin potkut. Myös Ballmer joutui lähtemään. Ei tällä materiaalilla kirjoiteta Troijaa uudelleen, korkeintaan muistelmia.

Elopin virheenä pidetään yleisesti Nokian intranetistä vuotanutta Burning Platform -kirjoitusta, jossa haukuttiin yhtiön tuotteet ja jonka väitetään tapanneen myynnin. Tämä on vasta osatotuus. Burning Platform vain runoili ongelmia ja ratkaisuvaihtoehtoja.

Suurin uhkapeli tapahtui Capital Markets Day -tilaisuudessa 11.2.2011 – päivämäärä sai hätänumerovertauksenkin. Siellä Windows Phonen kerrottiin korvaavan Meego- ja Symbian-tuotteet, lähes vuosi, globaalisti kaksi vuotta, ennen kuin nuo Windows Phonet olisivat kaupoissa. Elop perusteli avoimuutta muutoksen suuruudella.

Edellinenkin johto ei ollut syytön: vanhentunut Symbian oli pilkottu epäyhteensopiviin haaroihin ja alamäkeä oli tilkitty halpuuttamalla. Kun parempien Android-tuotteiden ekosysteemi kasvoi ja hinnat laskivat, Symbianin asema oli tukala. Nokian Linux-alustoja taas oli laiminlyöty aina, ja suunta androidiksi androidin paikalle hukattiin jo 2005.

Amerikkalaiset rakastavat supersankareita ja suuria eleitä, ja näin myös Elop jätti ristiriitaisimman perintönsä panemalla kaiken kerralla likoon. Jos itse julistus ei hidastanut myyntiä, sen teki viimeistään operaattoreiden kiinnostuksen loppahtaminen. Lisäksi tuolloinen Windows Phone 7 ei sopinut Nokian suurimmille markkinoille eikä tukenut Windows Phone 8:n sovelluksia.

Näin Nokia haaskasi vuodet 2011 ja 2012 odotteluun. Jäi kysymys, olisiko tulos parantunut, jos muutokset olisi salattu pidempään ja porrastettu viisaammin – tai hyödynnetty aikaisemmin Androidia.

Jokapäiväiset seikkailut

Niinpä Nokiaa ja Microsoftia luotsaavat nyt uudet, yhteensattumana intialaistautaiset toimitusjohtajat: Rajeev Surin Nokia keskittyy matkapuhelinverkkoihin ostamalla Alcatel-Lucentin ja myymällä Here-paikkatietoyksikön saksalaisille autojäteille. Satya Nadellin Microsoft on siirtänyt katseensa pilveen ja tulevaisuuden alustoihin, esimerkkinä Hololens-virtuaalilasit.

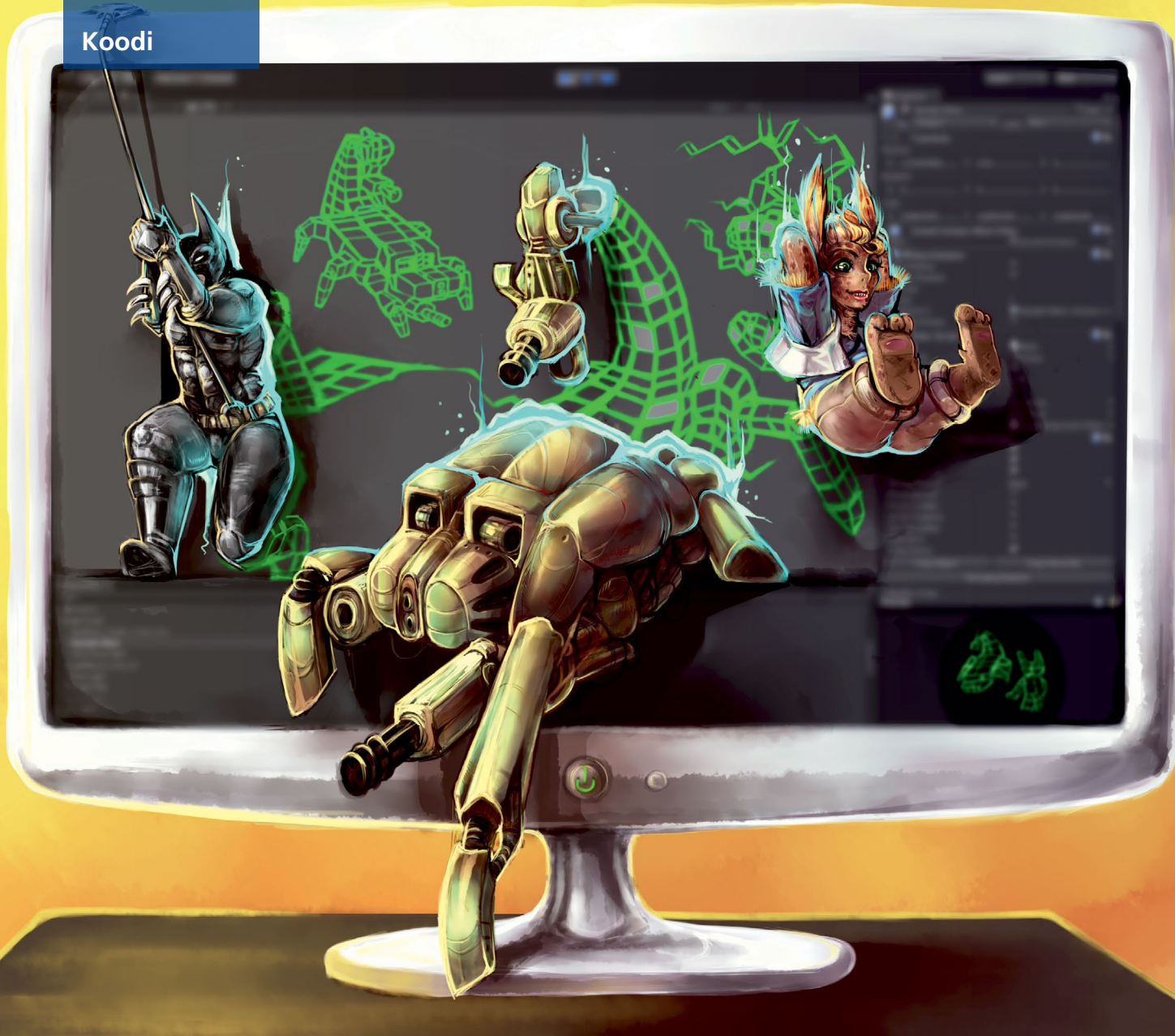
Microsoft on alaskirjannut ja sulauttanut kännykkähankintansa sekä julkaissut avainsovelluksensa kilpailuille mobiilialustoille. Ei yhtiö silti Windows Phonesta luovu: se fuusioituu Windows 10:een. Jos Windows 10 menestyy, pärjäävät puhelimetkin sen verran, että Microsoftille kelpaa. Tusinamalliston valmistus kuitenkin loppuu, ja tilalle tulee tiukka kolmen Lumian valikoima.

Nokiaankin kuuluu edelleen mystinen teknologiayksikkö, joka julkaisi kesällä elokuvastudioille tarkoitetun Ozo-virtuaalikameran. Nokia intoutui myös vahvistamaan ryhtyvänsä matkapuhelimiin, aikaisintaan sitten, kun kilpailukielto Microsoftin kanssa raukeaa loppuvuodesta 2016. Vanhaan ei kuitenkaan ole paluuta, ja Nokia puhuikin nyt brändilisensoinnista.

Luvassa on silti enemmän kuin pelkkää nimen myymistä. Nokia N1 -tablettimaistainen lienee osuva kuvaus uudesta mobiilitaktiikasta: brändi, design ja vähän koodia tulee Nokialta, Android-käyttöjärjestelmä Googlelta ja toteutus – siis valmistus, markkinointi ja jakelu – kumppanilta. Nokia N1 kertoo toki myös rajallisesta levityksestä: sitä sai pitkään vain Kiinasta.

Nokialla on muitakin tuotesuunnitelmia. Viime aikoina yhtiö on rekrytoinut ahkerasti työntekijöitä puettavan ja sensoriteknologian saralla. Suunnitelmia on lämmitelty mm. Here Everyday Adventures -merkin alla, eivätkä kaikki projektit suinkaan lähde karttayksikön mukana.

Nähtäväksi jää, kuoriutuuko Nokian seikkailuista strategiakin. 🐼



Pelinkehityksen vallankumous: ammattityökaluja ilmaiseksi

Mitä tarvitaankaan siihen, että saa tehtyä videopelin? Aluksi tarvitaan tietenkin inspiraatiota. Ei peliä voi tehdä, ellei ole ideoita siitä, mitä haluaa tehdä. Mutta ei pelkällä idealla vielä pitkälle pötkitä, vaan tarvitaan tietenkin myös taitoa toteuttaa se. Toisin kuin monet muut harrastukset, pelinkehitys on myös välineurheilua, sillä jotta yleensäkin voisi olla mitään pelattavaa, tarvitaan pelimoottori. Se juttu, joka piirtää grafiikan, soittaa äänet, käsittelee tiedostot, vastaanottaa ohjauksen ja pyörittää pelilogiikkaa.

Teksti: Miikka Lehtonen Kuvat: Outi Sinisalo, Miikka Lehtonen

Lukemattomat peliprojektit ovat tyssäneet siihen, että yritetään taistella huonosti dokumentoituja, kankeita tai muuten vain puutteellisia työkaluja vastaan ja sitten kyllästyään.

Onneksi nykyään kaikki on niin paljon helpompaa. Hyviä ammattitason työkaluja saa netistä ilmaiseksi niin paljon, kuin jaksaa imutella. Aina ei kuitenkaan ollut näin. Ei tarvitse mennä montakaan vuotta taaksepäin, kun

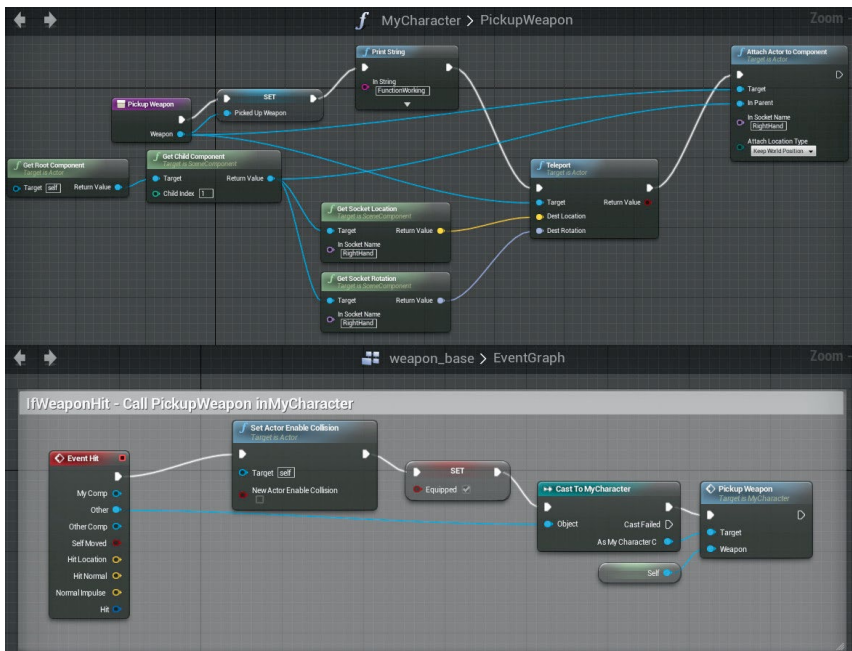
suurin osa nykyisistä kehitystyökaluista oli vielä pöytälaatikkokoodarien ulottumattomissa.

Tokihan *Unreal Enginen*, *CryEnginen* ja *Unityn* kaltaisilla moottoreilla on tehty pelejä jo vuosikausien ajan, mutta vielä muutamaa vuotta sitten pelinkehitysparatiisin portilla oli turhan kallis hintalappu. Itsekin juuri ennen suurten moottorien vallankumousta pelinkehitysharrastukseni aloittaneena muistan, miten Python-ohjelmointikielen varaan rakennettu *PyGame* ja

muut tuon ajan ilmaiset vaihtoehdot olivat todella hankalasti lähestyttävissä ja ominaisuuksiltaan rajallisia. Olisihan niilläkin varmasti saanut tulosta aikaan, jos olisi ollut paljon kärsivällisempi ja jaksanut rakentaa itse ne tarvittavat työkalut.

Vuoden 2009 ihme

Ennen vuotta 2009 amatööripelinkkehittäjien vaihtoehdot olivat todellakin aika karut. Tarjolla oli toki ilmaisia vaihtoehtoja, kuten jo mainittu



Unreal Enginen Blueprint-näkymä tekee pelikoodin visualisoinnista ja muokkaamisesta helppoa puuhaa. Näin vaikka kenttien suunnittelu on aiempaa sujuvampaa.

Pelimoottorien historia, lyhyt oppimäärä

Ennen Unreal Enginen yleistymistä elettiin pelinkehityksessä aikamoisessa villissä lännessä, jossa lähes jokainen firma ja tiimi käytti omaa pelimoottoriaan ja omia kehitystyökalujaan. Toki oli olemassa myös lisensointiin tarkoitettuja moottoreita, mutta ne olivat laadultaan hyvin vaihtelevia. Yhdessä ääripäässä on id:n *Doom*- ja *Quake*-moottorien kaltaisia hittejä, jotka olivat suosittuja sen takia, että ne olivat aikansa huipputasoa. Monet tiimit käyttivät nykyäänkin pelimoottoreita, jotka rakentuvat osittain *Quake*-moottorin varaan.

Pääosin kuitenkin tiimit tekivät itse moottorinsa ja työkalunsa, mikä oli varmasti ihan kivaa, jos sattui olemaan John Carmackin kaltainen nero, mutta vähemmän kivaa, jos halusi vain tehdä pelejä. Unreal Enginestä tuli vuosiksi alan standardi juuri sen takia, että se oli hyvin dokumentoitu ja yksinkertaisesti paljon parempi, kuin mihin monet tiimit olisivat itseksensä pystyneet.

Nykyään Unreal Enginen valta-asema on historiaa, mutta peliala on muuttunut. Enää muut vaihtoehdot eivät ole niinkään yksittäisten tiimien omia moottoreita, vaan kaikkien Electronic Artsin studioiden yhteisvaikutuksella rakennettuja ratkaisuja, joissa monista vahvoista palasista muodostuu erinomainen kokonaisuus.

PyGame tai *Adventure Game Studio*. Muutamat rohkeat uskalsivat veloittaa rahaa omista kehitysympäristöistäänkin. Esimerkiksi *GameMakerin* avulla on todistettavasti tehty kivoja ja näppäriä 2D-pelejä, esimerkiksi alkupe räjähen *Spelunky*.

Teoriassa mikään ei estänyt amatöörejäkään hankkimasta ammattitason työkaluja, kuten vaikka Unity 3D:tä, Unreal Engineä tai CryEngineä, joilla kaikilla oli tehty todella nättiä ja kovatasoista jälkeä. Ainoa ongelma oli se, että moottoreista pyydettiin kovia summia. Halvimmatkin vaihtoehdot maksoivat joitain tuhansia, mikä on todella kova este satunnaiselle ja vähän puolivakavammallekin indie-pelikehitykselle.

Syksyllä 2009 alkoi vallankumous, joka on pitkälti mahdollistanut nykyisen indie-kehitysskenen muodostumisen, sillä nämä kovat ammattitason työkalut muuttuivat jos nyt eivät täysin ilmaiseksi niin ainakin helpommin saataviksi. Ensin ehti Unity Technologies, joka julkaisi Unity 3D:stä rajoituneen, mutta ilmaisen indie-version. Unity saikin nopeasti paljon suosiota ja näkyvyyttä, mikä oli varmasti osatekijä, kun marraskuussa 2009 paukahti todellinen pommi. Elettiin PlayStation 3:n ja Xbox 360:n kulta-aikaa, ja pelimoottorien kiistaton kuningas oli Unreal Engine. Joskus ammoisina aikoina miltei jokaisella pelistudiolla oli oma moottorinsa, jota sitten vartioitiin

Kenttäeditorit ja modajaat, alan pioneirit

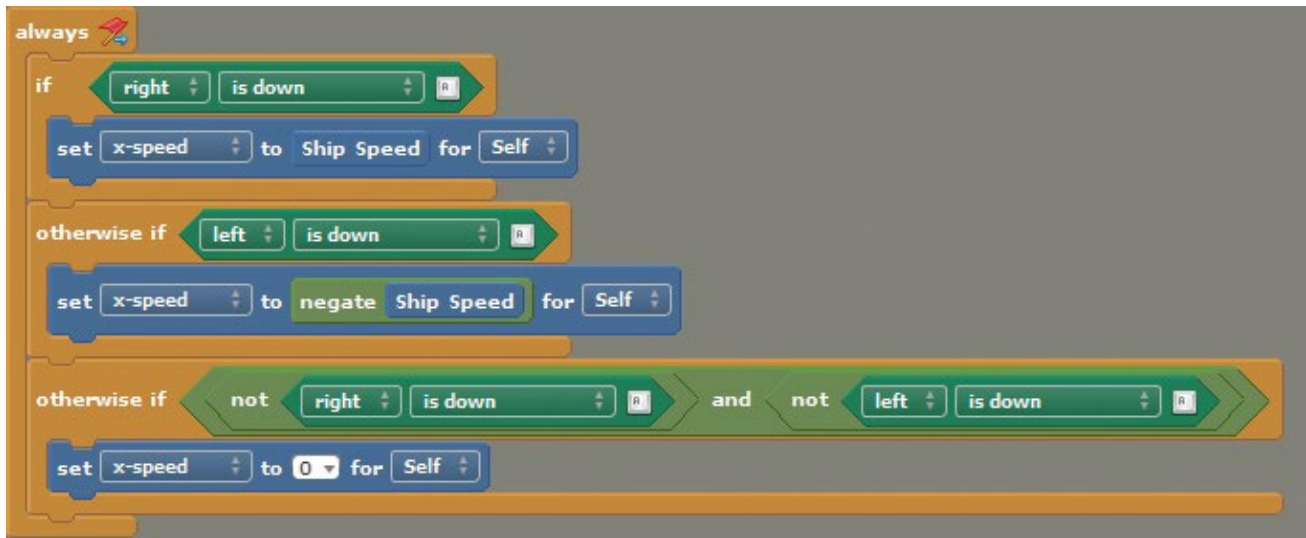
Indie-pelikehityksen juurista puhuttaessa täytyy muistaa myös erilaiset modit ja kenttäeditorit. Moni nykyään ammatikseen pelejä kehittävä on saanut alkunsa tekemällä vaikka kenttiä *Doomiin* tai kehittämällä modeja peleihin. Kun on valmis runko ja hyvin dokumentoidut työkalut, kuten vaikka Valven *Source*-pelimoottorin tai id:n *idTech*-pelimoottorin tapauksessa perinteisesti on ollut, innokkaat amatööri pystyvät loittimaan olemassa olevien pelien kylkeen tai päälle hämmäntävän erilaisia ja itsenäisiä peliteoksia. Esimerkiksi megasuositun *DOTA*-peli sai alkunsa *Warcraft III*-modina, kun taas *Counter-Strike* oli alkujaan *Half-Life*-modi.

karhuemmon innolla muiden lähestymisiltä. 2000-luvun alkuvuosina trendi kuitenkin muuttui ja Unreal Enginestä tuli se oletusarvoinen valinta miltei kaikissa vähänkään suuremmissa peliprojekteissa. Niinpä oli melkoinen yllätys, kun Unreal Engineä kehittävä Epic Games julkaisi moottoristaan ilmaisen version.

Noiden kuukausien kait ja seuraukset tuntuvat ja näkyvät nykyäänkin pelinkehityksen maailmassa. Ensinnäkin pienten tiimien määrä on moninkertaistunut räjähdysmäisesti, eikä ihme. Kun ammattitason työkaluja on saatavilla ilmaiseksi tai pilkkahintaan, kynnyksen kokeilla pelinkehitystä ja huomata sen olevan kivaa puuhaa on matalampi kuin koskaan. Samaan aikaan Steamien tapaisten digitaalisten jakeluverkostojen yleistyminen on mahdollistanut pelien itsejulkaisun ja -jakelun. Ei sovi myöskään unohtaa mobiilipelaamisen nousua, sillä juuri mobiilialustat ovat perinteisesti olleet Unityn kotikenttä.

Kun käyttäjäkannat ovat kasvaneet ja indie-peleissä liikkuu aina vain enemmän ja enemmän rahaa, myös moottorialalla on alkanut todellinen kilpavarustelu. Varsinkin Unity ja Unreal Engine kehittyvät kovaa vauhtia.

Nyt kun olemme kerranneet pikavauhtia modernin pelinkehityksen historian, tutustutaanpa tarkemmin niihin suuriin tekijöihin ja niiden mahdollisuuksiin – sekä myös mahdollisiin ongelmiin.



Stencyl on hyvin helppo tapa aloittaa pelinkehitys. Pelilogiikka suunnitellaan tavalla, joka on välittömästi ymmärrettävä ja intuitiivinen. Näin pääsee asiaan, vaikka koodaus ei ihan sujuisikaan. Tässä tehdään avaruusalukselle sivuttaisohjausta.

Minkä sitten valitsisin?

Mikä sitten on se pelimoottoreista ja kehitysympäristöistä muikein? Jos on täysi noviisi, Stencyl on oikein mainio paikka aloitella, sillä sen avulla pääsee nopeasti käsiksi siihen hauskaan juttuun, eli tekemään peliä. Jos taas bitti pysyy hyppysissä ja koodi sujuu, ihan rehellisesti sanoen tekisin valintani hallussa olevien ohjelmointikielten perusteella. Vaikka moottoreissa ja kehitysympäristöissä on eroja, suuret linjaukset ovat niin samanlaiset, että molemmilla pääsee hyvin alkuun.

Jos C# tai JavaScript ovat lähellä sydäntä, vastaus on Unity. Jos taas C++ sujuu, kokeile Unreal Engineä.

Unity

Unity Technologiesin Unity 3D oli todella pitkään indie-kehityksen kiistaton kuningas. Osittain taustalla oli puhtaasti se, että Unity oli ensimmäinen suurista ilmaisimoottoreista. Se nappasi merkittävän osan indie-kehittäjien huomiosta ja tämä kierre ruokki lisää huomiota, kun Unityä suositeltiin helposti muillekin innokkaille yrittäjille.

Vähintään yhtä merkittävää oli kuitenkin se, että toisin kuin monet muut pelinkehitysympäristöt ja -moottorit, Unity oli suunniteltu helposti lähestyttäväksi ja miellyttäväksi käyttää. Unity tuli tunnetuksi siitä, että sen avulla sai todella helposti ja nopeasti toimivaa kamaa ruudulle. Ei silmänräpäyksessä tietenkään mitään viimeisteltyä peliä tehty, mutta aikana jolloin yksinkertaisenkin idean testailu vaati normaalisti paljon pohjustusta ja työtä, Unity tuntui todellakin vallankumoukselliselta.

Vallankumouksen keskipisteessä oli Unityn paras idea, Scene Editor. Nor-

maalisti pelejä kehittäessä kentät tai vaikka seikkailupelin huoneet täytyi joko koodata käsin sijoittamalla esineitä koodin avulla tiettyihin koordinaatteihin tai sitten toteuttaa itse jonkinlainen kenttäeditori, jolla saman homman saisi tehtyä mukavammin. Unityssä tämä ominaisuus tuli vakiona mukana. Oli suorastaan mieltäräjäyttävää, kun pystyi raahailemaan 3D-esineitä pitkin kolmiulotteista avaruutta, painamaan play-nappia ja sitten olemaan keskellä juuri rakentamaansa huonetta vain minuuttien työn jälkeen. Scene Editorin avulla esineisiin ja objekteihin pystyy myös drag & drop-periaatteella liittämään jos jonkinlaisia komponentteja: koodinpätkiä, äänitiedostoja, animaatioita ja niiden soittamiseen tarkoitettuja häpäkkeitä.

Unity on myös koodarin näkökulmasta erinomaisen mukava käyttää. Tuettuja kieliä ovat JavaScript, C# ja muistona muinaisilta ajoilta oliopohjainen harvinaisuus, Boo. Unityn omat laajennukset ovat hyvin dokumentoituja ja miellyttäviä käyttää. Jos kuitenkin tarvitsee itse jotain, Unity laajenee kivasti tarvittaessa vaikka ihan raaioilla C#-luokilla.

Kuten arvata saattaa, helppo laajennettavuus ja kotikoodariskene ovat yhdessä luoneet Unityn ympärille todellisen laajennusparatiisin. Jos Unityn editori tai taustalla rullaava pelimoottori eivät valmiiksi tue jotain ominaisuutta, todennäköisesti joku innokas kotikoodari on sellaisen jo värkännyt ja joko jakelee sitä ilmaiseksi tai myy sitä Unityn kylkeen pultatun laajennuskaupan välityksellä.

Erinomainen esimerkki tästä on vaikka ajurituki. Unity itse kyllä teoriassa tukee peliohjaimia, mutta ei hirveän hyvin. Saatavilla on kuitenkin erinomaisen

mainio plugin, joka mahdollistaa käytännössä kaikkien DirectInput- ja Xinput-rajapinnat täyttävien peliohjainten käytön peleissä. Tai miten olisi 2D-tuki? Alkujaanhan Unity 3D kehitettiin vain 3D-pelejä varten, mutta ei mennyt aikaakaan ennen kuin suomalainen Juha Kiili oli jo kyhännyt mainiot 2D-työkalut editoriin ja moottoriin. Nykyään mies sitten työskenteleekin Unityllä.

Unityn suosiota on edesauttanut myös se, että kun pelinsä on kerran tehnyt vaikka Windowsille tai Androidille, sen voi verrattain helposti portata muillekin alustoille. Ihan mistään ”valitse alusta ja paina nappia” -tason porttauksesta ei kuitenkaan ole kyse. Ensinnäkin kaikkia kohdealustoja ei ole ihan noin vain kaiken kansan saatavilla. Moderneille pelikonsoleille ei esimerkiksi voi pelejä kääntää, ellei ole hyväksytty ja lisensoitu kehittäjä.

Lisäksi alustoilla on omat erikoisuuksensa. Olen itsekin tapellut usein pienten outouksien kanssa: Androidilla 2D-valikkografiikat toimivat ihan mainiosti, iOS:llä kaikki pusertuu neliöksi. Syytä etsittiin kissojen ja koirien kanssa, kunnes kävi ilmi, että kyse oli iOS:n käyttämän OpenGL-version erikoisuudesta.

Kokonaisuutena Unity on kuitenkin erinomainen pelinkehitysympäristö ja moderni pelimoottori. Vielä muutama vuosi sitten sitä ei olisi voinut suositella ihan kaikenlaisiin peleihin, mutta nykyään on toisin, sillä Unity on kehittynyt niin paljon, että vain pelinkehittäjän oma mielikuvitus on rajana.

Unreal Engine

Osasyy sille, että Unity on vieläkin käytännössä automaattinen valinta



Aivan tällaiseen jälkeen amatööriinit eivät ehkä pysty, mutta tuore Batman: Arkham Knight on hyvä esimerkki siitä, millaista jälkeä Unreal Engine 4:llä saa aikaan.

mobiili- ja indiekehityksessä löytyy Unreal Enginen ja Unityn filosofisista näkemyseroista. Siinä missä Unity kehitettiin alkujaan pienten porukoiden ja suoraan sanoen myös pienten pelien ympärille, Unreal Engine suunniteltiin suuria AAA-luokan pelejä varten. Se näkyi vähän joka puolelta.

Sitten Epic kuitenkin ajoi seinään. Unreal Enginen yksinvalta AAA-pelinkehityksessä murtui suurten julkaisijoiden siirtyessä taas omien moottoriensa pariin ja samalla indiet eivät kaivanneet Unrealin skaalaa tai kaikkia sen suurimpiin produktioihin tarkoitettuja ominaisuuksia. Niinpä Epic muuttui aikojen myötä ja tuunasi Unreal Engineä joustavammaksi. Nykyään se onkin erinomainen pelimoottori ja -kehitysympäristö jos jonkinlaisiin peleihin.

Unityn tavoin myös Unreal Engine on rakennettu toimivan, joustavan ja monipuolisen editorin ympärille. Editorissa pystyy kätevästi rakentelemaan pelimaailmoja, tuunaamaan animaatioita ja niin edelleen. Pienet filosofiset erot kuitenkin paistavat yhä läpi raosta ja kielivät siitä, että Unreal Engineä kehitetään vähän suurempien studioiden tarpeet mielessä.

Moottorin ehkä mielenkiintoisin ominaisuus on nimittäin Blueprint, joka on eräänlainen visuaalinen skriptaustryökalu. Blueprint-näkymässä peliprojektin eri luokat näkyvät graafisina esityksinä ja niitä voi sitten yhdistellä

toisiinsa pelilogiikan avulla. Luokkien julkiset liitännät ovat kätevästi näkyvillä, ja homma toimii pienen opetteluun jälkeen kuin tanssi.

Koodarit voivat toki pohtia, paljonko tuostakin sitten on hyötyä, mutta ominaisuutta ei olekaan kehitetty ensisijaisesti meitä varten, vaan produktioihin, joissa myös ei-koodarien pitäisi päästä käsiksi koodiin ymmärrettävällä tavalla. Kuvitellaan vaikka suuri peli, jossa on erikseen pelikenttien suunnittelijat. On todella kätevää, kun ei tarvita mitään erillistä kenttäeditoria, vaan pelisuunnittelijakin voivat itse toteuttaa pelilogiikkaa ja suunnitella kenttiä ja kohtauksia helposti ja joustavasti.

Ei kannata kuitenkaan antaa tällaisen juttujen pelotella, sillä vaikka Unreal Engine onkin ehkä suunniteltu vähän kehittyneempään käyttöön kuin Unity, molempia voi hyvin käyttää ristiin: siinä missä Unityllä onnistuvat suuretkin produktiot, Unrealinkin oppii. Toki pienoisena kynnyksenä nykyään voi olla se, että siinä missä Unity tukee useampia ohjelmointikieliä, Unreal Engine ymmärtää vain C++:aa. Toki tämäkin on parannusta aiempaan, sillä männävuosina Unreal Engine pyöri UnrealScriptin varassa. Se taas oli Epicin itse kehittämä oliopohjainen kieli, joka oli kyllä ihan toimiva, mutta yksittäisten pelimoottorien omat kielet ovat aina vähän rasittavia.

Yhtäläisyyksiä selittää paljon se, että jos jompikumpi suurista mootto-

rivalmistajista kehittää jotain uutta ja vallankumouksellista, toinen yleensä kopioi sen melko pian omaankin kehitysympäristöönsä. Niinpä Unreal Enginestäkin löytyy oma kauppapaikkansa, josta voi Unityn tavoin ostella koodia, grafiikkaa, ääniä ja muuta muikkea.

Unrealin suurten tuotantojen jäljet näkyvät myös siinä, että se tukee oletusarvoisesti aivan eri tason silmäkarkkia kuin vaikka Unity. Unreal Engine on suunniteltu ottamaan kaikki väantö irti modernistakin raudasta ja täten amatöörimäisinkin indie-kehittäjä saa nappia painamalla peliinsä sellaista silmäkarkkia, että oksat pois.

Tämä on joskus myös ongelmallista,

Mitä niillä on tehty?

Pelimoottorit ja kehitysympäristöt ovat sinänsä vähän abstrakti juttu, joten annetaanpa niille tavallaan kasvot. Näitä pelejä erilaisilla moottoreilla ja ympäristöillä on saatu aikaan.

- **Unreal Engine:** muun muassa *Gears of War* -sarja, *Batman*-pelit, *Borderlands*-sarja
- **Unity:** *Pillars of Eternity*, *Kerbal Space Program*, *Cities: Skylines*, *Crossy Road*
- **Source:** *DOTA 2*, *Half-Life 2*, *Team Fortress 2*
- **Stencyl:** *Lakeview Cabin*, *Spiderling*, *Tiny Dangerous Dungeons*

VR? OK!

VR eli virtuaalidellisuus on pelivuoden 2015 kuuma puheenaihe. Valve, HTC, Oculus, Sony ja lukemattomat muut firmat ovat puske massaa ulos omia lasejaan. Aihe kiinnostaa paitsi pelaajia, myös pelinkehittäjiä, sillä VR tarjoaa aivan omanlaistaan tunnelmaa ja vaatii erityishuomiota kehittäjiltä.

Ilahduttavasti kehitysympäristöt ja pelimoottorit alkavat olla VR-valmiita. Tuoreen Unreal Engine 4.8:n myötä Unreal Engine sisältää natiivin tuen kaikille merkittävillä laseilla, myös Sony:n Project Morpheukselle. Unitykään ei jää paljon pekkää pahemmaksi, sillä senkin tuorein versio lisäsi yhtenäisen VR-ratkaisun. Nyt homma ainakin teoriassa toimii niin, ettei kehittäjän tarvitse juuri huolehtia siitä, mitkä lasit käyttäjällä on.

sillä kun ominaisuuksia ei tarvitse kehittää itse tai edes ymmärtää millään tasolla, ei aina myöskään ymmärretä sitä, miten paljon resursseja ne vaativat. Onkin aika arkipäiväistä törmätä Unreal Enginellä tehtyihin peleihin, joiden grafiikkasuunnittelussa ohjenuorana on ollut ”kaikki grafiikat, kiitos”. Sitten voidaankin pohtia, millainen tietokoneiden vastike Graalin maljalle kotoa pitäisi löytyä, jotta pelit pyörivät pehmeästi.

Ja entäs ne muut?

Ehkä olet maailman indiein indiekehittäjä, etkä halua koskea mihinkään vähänkään kaupalliseen. Tai ehkä haluat vain jotain vielä helpompaa, tai vain puhtaasti erilaisuutta erilaisuuden nimissä. Se on ihan OK. Vaihtoehtoja löytyy.

Pitänee toki mainita *CryEngine*, saksalaisen CryTekin pelimoottori, jolla pystyy tekemään tautisen kaunista jälkeä moderneille alustoille. Itse en ole kuitenkaan siihen sen tarkemmin tutustunut, joten en uskalla sen hyvistä tai huonoista puolista juuri mainita.

Huomattavasti mielenkiintoisempi vaihtoehto on *Stencyl*, joka on ensisijaisesti tarkoitettu ihmisille, jotka eivät osaa koodata. Ensimmäisellä mieleen tulee vahvasti legendaarinen 90-luvun alun *Klik n Play*. *Stencyl* on toki monipuolisempi, mutta molemmat ohjelmat pyrkivät tekemään algoritmista ajattelusta ja koodin suunnittelusta visuaalista ja helposti ymmärrettävää puuhaa. Esineisiin ja olentoihin voi liittää logiikkaa, jota tehdään sitten helppojen if-then-mekaniikkojen avulla.

Homma toimii, sillä *Stencyl* on tarpeeksi monipuolinen ja joustava mahdollistamaan jos jonkinlaisia pelejä. Itse en ole sen kanssa hirveästi pelejä kehitellyt, mutta se vaikuttaa erinomaisen toimivalta ja kätevältä paikalta aloittaa pelinkehityksen opettelu. Oh-

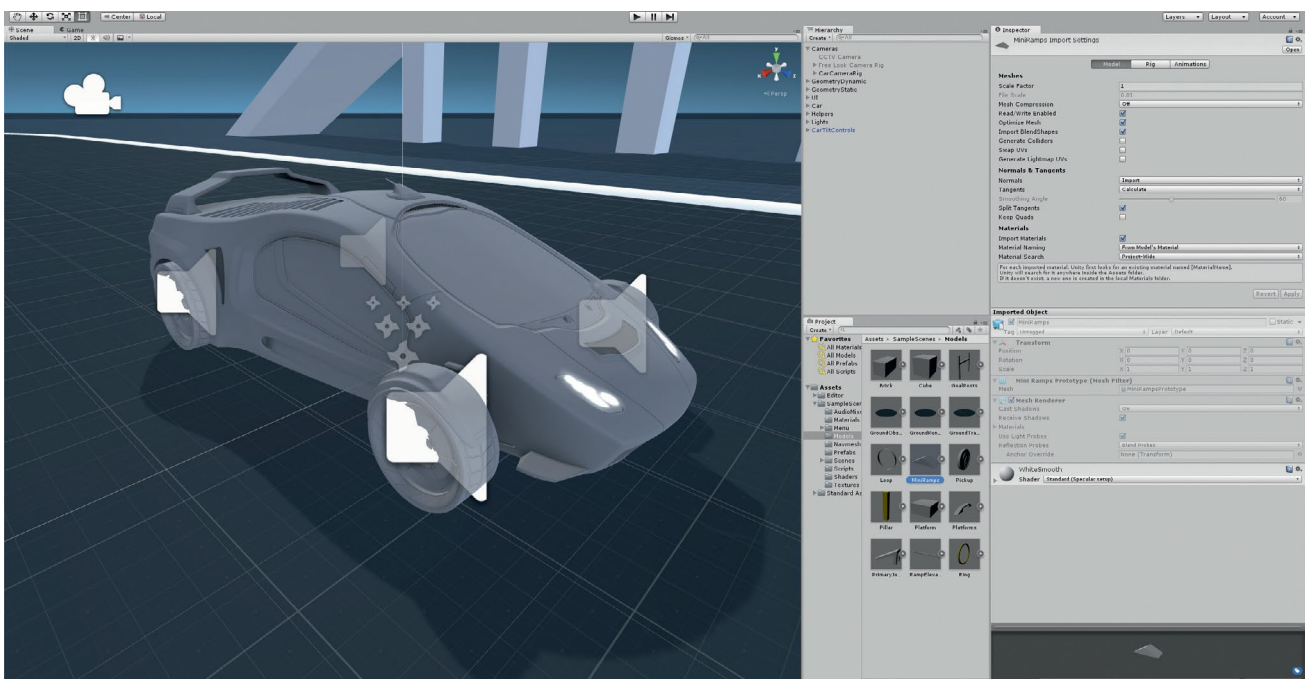
jelmoinnissahan tunnetusti vaikeinta on nimenomaan algoritmisen ajattelu: miten otetaan jokin konkreettinen toimi tai asia ja puserretaan se sellaisiin loogisiin askelmiin, että tuloksena on koodia. Loppu on vain semantiikkaa ja sanavaraston kasvattamista.

Aloitti millä moottorilla aloitti, pelinkehitys on suositeltava harrastus. Vaikka tuloksena ei olisi koskaan yhtään katselukelpoista peliä, yksinkertaistenkin projektien läpivienti opettaa ja antaa paljon. 🍷

Mistä sitten rahat?

Kun moottoreita levitetään ilmaiseksi, mistä rahat sitten tulevat? Osittain rojalteista eli prosenttiosuudesta, joka maksetaan moottorin kehittäjälle. Unityn tapauksessa alle 100 000 taalaa tuottavista peleistä ei tarvitse maksaa mitään, Unreal Engin tapauksessa taas 5% tuloista menee Epicille, jos teos tuottaa yli 3000 taalaa vuosineljännekseltä.

Lisäksi myynnissä on ammattilaisversioita moottoreista ja kehitysokaluista. Aiemmin maksun taakse oli lukittu perustason ominaisuuksia, mutta nykyään ammattilaiset saavat lähinnä ammattilaisten tarvitsemia juttuja: paremmat tiimityökalut, yksilöityä tukea ja niin edelleen.



Unityn Scene Editor tekee pelimaailman suunnittelusta ja assetien käsittelystä todella sujuvaa. Aiemmin tämä kaikki olisi vaatinut omien työkalujen koodaamisen, mikä on parhaimmillaankin hankalaa.

Commodore 64: a visual Compendium

230+ sivua, väripainatus.

Julkaisija: Funstock.co.uk**Tekijä:** Bitmap Books, Sam Dyer**Saatavuus:** www.funstock.co.uk, hinta 35 euroa + postikulut. PDF-kirja 10 puntaa.

Sam Dyerin C64-taidekirja oli herran ensimmäinen Kickstarterin kautta joukkorahoitettu pelitaideteos. Kirja on selvästi ohuempi kuin myöhempi Amiga-teos, mutta se ei sen kauneutta syö. Kuusnelosen pikselit kun ovat niitä kaikkein kauneimpia ja sympaattisimpia.

Kuusneloskirjan sympaattisimpia hetkiä ovat ne, kun sivua kääntäessään eteen tulee jotain täysin odottamatonta. Kuten vaikkapa Crash- ja ZZAP 64 -lehdissä vaikuttaneen taitelijan Oliver Freyn uskomattomia kansikuvia tai julisteita. Niiden parissa kasvoi kokonainen pelaajasukupolvi.

Samalla tulee selväksi peligrafiikan hurja kehittyminen 1980-luvun mittaan. C64:n rauta ei muuttunut, mutta sekä taitelijat

että kooderit oppivat huikeita uusia kikkoja. Niiden myötä sotaratsusta revittiin irti upeita suorituksia aina 3D-grafiikasta todella viimeisen päälle viilattuun pikselitaitteeseen. Aina ei uskoisi, että kaikki nämä pelit on tehty samalle koneelle.

Ja onhan se retroajalle herkkää, kun moni aikoinaan rakas peli muistuu mieleen ja herättää kauan sitten hautautuneita muistoja. Tämä kirja kuuluu commodoristin kirjahyllyyn, siitä ei pääse mihinkään.

Jukka O. Kauppinen



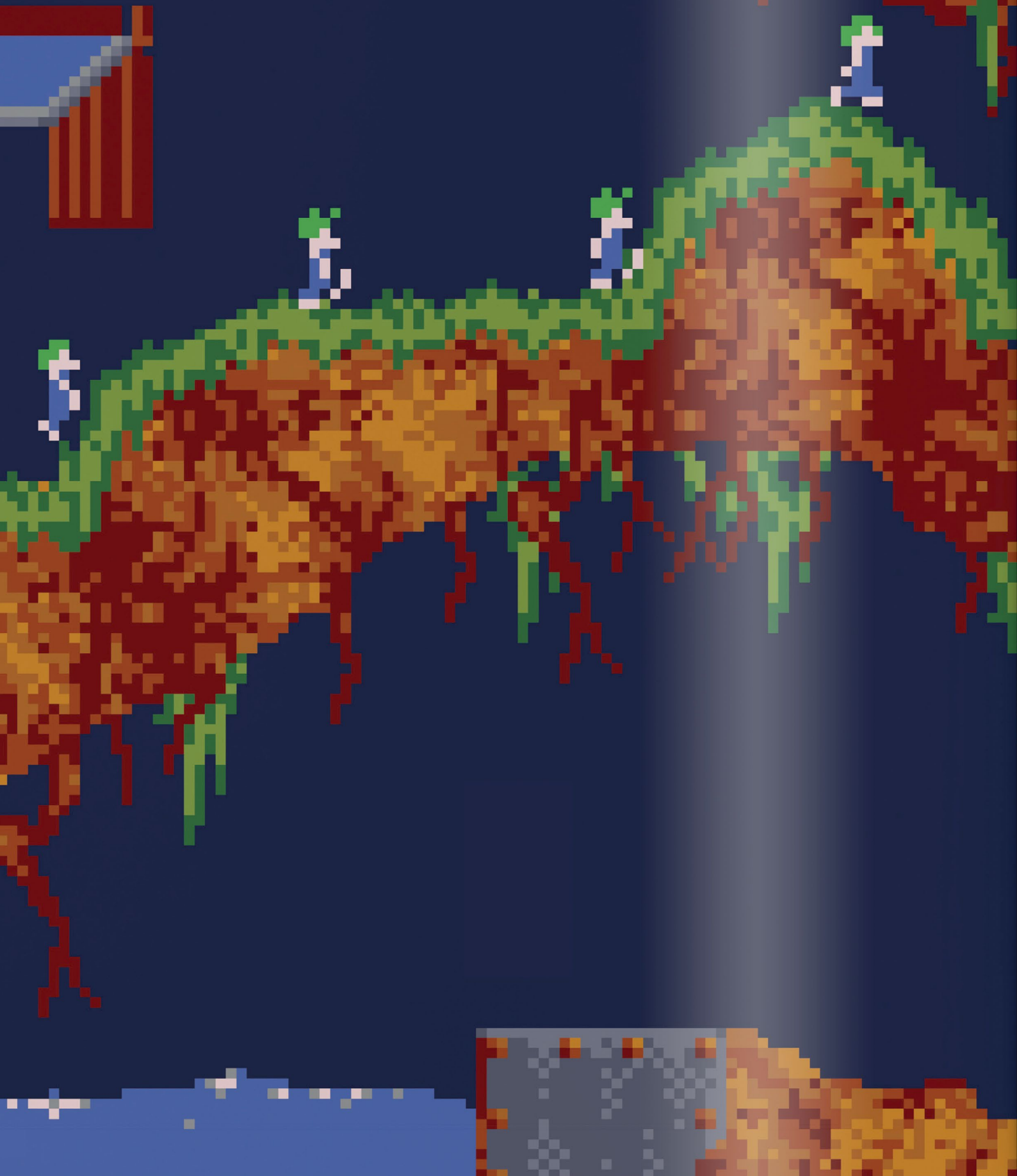
Commodore Amiga: a visual Compendium


426 sivua, väripainatus.

Julkaisija: Funstock.co.uk

Tekijä:Bitmap Books, Sam Dyer

Saatavuus: www.funstock.co.uk, hinta 29,99 puntaa + postit





Miten kaunis onkaan pikseli? Joskus verrattoman kaunis. Ja kun Amigan pikseleitä painetaan paperille ja levitetään yli 400 sivulle, on tulos muistoja ja ajatuksia nostattava, hyvin erilainen historiategos.

Commodore Amiga: a visual Compendium ei ole tietoteos, vaan visuaalinen läpileikkaus Amigan historiaan ja sen pelien taiteeseen ja siihen, mitä ne meille viestivät ja tarkoittavat. Mitä muistoja ne herättävät? Teos ei niinkään kerro vaan pikemminkin kysyy: mitä nämä kuvat ja pienet tarinat sinulle merkitsevät? Mitä ajattelet katsoessasi aukeamalle levitettyjä kuvakaappauksia?

On myös ihailtava kirjan toimittajien rohkeutta esittää pikseli rohkeasti omana itsenään, paljaana ja alastomana.

Teos on matka siitä, mistä kaikki alkoi ja miksi Amigaa rakastettiin, aina sen viimeisiin päiviin saakka – ja kauemmas. Ja missä me kävimme tämän pitkän, upean matkan aikana. Se on myös kirja, josta olisi hauska lukea jonkun täysin ulkopuolisen kanssa. Mitä ajatuksia nämä historiallisten pelien kuvat ja kuvaukset herättävät nykynuorissa? Riittääkö 320×256 enää mihinkään?

Jukka O. Kauppinen



Pelejä pelastamassa – Haastattelussa videopeliarkiston katastrofisuunnittelija

Digitaalisen aineiston säilyttämiskäytännöt ja arkistointi ovat viime aikoina Suomessakin esille nousseita kysymyksiä. Miten michiganilaisen yliopiston videopeli- ja tietokonearkistoa ylläpidetään, hoidetaan ja suojellaan? Miten arkistossa on varauduttu kriisitilanteisiin?

Teksti: Mikko Tuomela

Kuvat: Mikko Tuomela, Michiganin yliopisto, David Carter

Opiskellessani University of Michiganissa vuosina 2010–2012 tutustuin Rebecca Frankiin, joka oli töissä yliopiston kirjaston ylläpitämässä Computer & Video Game Archivessa (CVGA). Se on kampuksella sijaitseva, kaikille avoin peliarkisto, joka arkistoi ja säilyttää pelejä ja pelikoneita ja jossa pelejä voi myös pelata ja käyttää osana tutkimusta. Aloitettuaan työskentelynsä Rebecca Frank laati kirjaston arvokkaalle kokoelmille kattavan katastrofisuunnitelman ja tekee nykyään väitöskirjaa digitaalisesta säilyttämisestä. Kyselin hänen kokemuksiaan suunnitelman laatimisesta ja peliarkiston arjesta.

Peliarkiston suojelemiseen liittyy paljon uniikkeja ongelmia, joita ei ole kirjojen kanssa.

Miten kiinnostuit digitaalisten aineistojen säilytyksestä ja päädyit CVGA:lle?

Kyseessä oli enimmäkseen sattuma. Opiskelin Master of Science in Information (MSI) -tutkintoa varten University of Michiganiin kuuluvasa School of Informationissa (UMSI). Samalla työskentelin taiteen, arkkitehtuurin ja tekniikan kirjastossa (AAEL), jossa yliopiston tietokone- ja peliarkisto sijaitsee. Työskentelin pääasiassa käsikirjastossa, mutta minulla oli mahdollisuus osallistua myös muihin projekteihin kirjastossa. En tuntenut peliarkistoa ennestään mutta olin innoissani, kun pääsin tutustumaan siihen ja työskentelemään CVGA:ssa.

Mikä motivoi sinut kartoittamaan arkistojen katastrofisuunnittelua?

Olen aikaisemmin työskennellyt liikkeenjohdon konsultoinnissa ja yritysten tutkimushankkeissa, ja mitä enemmän olin siinä maailmassa, sitä enemmän minua alkoi kiinnostaa, miten digitaalisia aineistoja käsitellään. Tajusin, että jouduimme keksimään pyörän aina uudelleen kutakin projektia varten, koska aineistot eivät olleet muodossa, jossa niitä voisi hyödyntää. Sitä kautta kiinnostuin digitaalisesta säilyttämisestä ja asiakirjahallinnosta.

Päädyin opiskelemaan MSI-tutkintoa, koska siinä voin yhdistää ko-



Rebecca Frank.

kemukseni kirjastotyöstä ja kiinnostukseni ratkoa työssä kohtaamiani ongelmia. School of Informationissa huomasin, että kaikki digitaalisiin aineistoihin liittyvät ongelmat koskevat myös kulttuuriperintöalaa, ja sitä kautta kiinnostuin digitaalisesta säilyttämisestä tutkimusalana.

Työskentelin koko kirjastoa koskevassa projektissa, jossa kehitettiin katastrofisuunnitelmia, ja sitä kautta päädyin työskentelemään CVGA-peliarkistossa. Koska peliarkisto sijaitsee kotikirjastossani AAEL:ssä, arkisto ja sen henkilökunta olivat minulle tuttuja. Peliarkiston kokoelmatkin olivat tulleet tutuiksi, joten sain edistettyä peliarkiston katastrofisuunnitelmaa paljon. Onnistuin myös yhdistämään sen opintojeni kanssa ja tein CVGA:n säilytysarvioinnin samaan aikaan. Tämä yhdistelmä – arviointi, suositusraportti ja katastrofisuunnitelma – sai minut miettimään, miten peliarkistoa tulisi hoitaa.

Millaisista asioista katastrofisuunnittelu koostuu?

Peliarkiston katastrofisuunnitelma on itse asiassa vain osa koko kirjastorakennuksen suunnitelmasta. Koska arkisto on niin erityinen, sitä lähestyttiin itsenäisenä kokoelmana, mutta suunnitelman piti silti istua koko kirjaston suunnitelmaan. Meidän piti priorisoida CVGA:n eri osiot mutta samalla priorisoida se koko kirjaston sisällä.

Yleisesti ottaen katastrofisuunnitelma CVGA:lle tai sen kaltaiselle kokoelmalle sisältää tietoa siitä miten kriiseihin reagoidaan ja miten niistä toivutaan, henkilökunnan yhteystiedot, pelastusprioriteetit eri aineistoille,

ohjeet kokoelmien siirtämiseen ulos rakennuksesta, tiedot välineistä, joita tarvitaan onnettomuuden sattuessa, sekä tiedot siitä, miten kokoelman osista huolehditaan, jos ne vaurioituvat.

CVGA oli kiinnostava tapaus. Tein suojeluarviointiraportin vuosina 2010–2011 ja sen ansiosta sain käsityksen kokoelman kunnosta. Se sisältää pelit, konsolit, muut laitteistot kuten televisiot jne. Lisäksi kartoitettiin säilytystilat, tilankäyttö, tilojen rajoitteet ja sen sellaista. Kokoelma oli juuri muuttamassa rakennuksen sisällä toiseen paikkaan, joten otin myös uudet tilat huomioon.

Miten peliarkisto luokitellaan kirjastossa? Katsotaanko se erikoiskokoelmaksi?

Minusta sitä voidaan kutsua erikoiskokoelmaksi. Itse asiassa arkkitehtuurin ja tekniikan kirjastolla on muutamia erikoiskokoelma. Monilla kirjastoilla on niitä, mutta meidän peliarkistostamme tekee erityisen se, että tavoitteena on säilyttää myös pelaamisen kokemus, ei pelkkiä pelejä ja konsoleita. Koko CVGA:n sisältö on tarkoitettu käytettäväksi, ja tämä asetti vaatimuksia sille, miten kokoelmia tuli lähestyä. Videopelien tarkoitus on tulla pelatuksi, sen sijaan että ne pidettäisiin lukkojen takana, joten peliarkisto muistuttaa enemmän kirjastoa kuin arkistoa, paitsi että aineistot ovat käytettävissä vain paikan päällä eli niitä ei voi lainata kotiin.

Millaisia erityishuomioita tehtiin nimenomaan peliarkistoa varten?

Yksi merkittävimmistä seikoista oli sähkö. Häiriöt sähköverkossa vai-



kuttavat kaikkeen, ja ne voivat myös vaurioittaa kokoelmaa. Sähkökatko ei vaikuta tavallisiin kirjoihin mitenkään, mutta meillä on paljon vanhoja sähkölaitteistoja, jotka eivät välttämättä pidä sähkökatkoista.

Ongelmatilanteissa pitää kaikki laitteistot pystyä sammuttamaan nopeasti. Koska peliarkisto on kellarikerroksessa, pitää varautua vesivahinkoihin, sillä ne luovat aikamoisen vaaratilanteen. Peliarkistolla on huone täynnä elektroniikkaa, jossa on virta päällä, joten myös mahdolliset tulipalot vaativat erityishuomiota.

Arkiston suojeleminen on haastavaa mutta niin on myös henkilökunnan suojeleminen. Tämän vuoksi meidän piti ottaa huomioon muun muassa sähköpalot ja oikosulut. Suuri osa kokoelmista on kovassa käytössä, joten välillä ongelmana ovat piuhojen kärähtämiset. Kaiken kaikkiaan peliarkiston suojelemiseen liittyy paljon uniikkeja ongelmia, joita ei ole kirjojen kanssa.

Miten tulviin ja muihin vesivahinkoihin reagoidaan?

Tarkoituksena on saada kokoelmat ulos. Suurin osa kokoelmasta on pyörillä varustetuissa kaapistoissa, joten jos vain on aikaa ja riittävän turvallista, niin kärkyt haetaan ulos, ovet suljetaan ja henkilökunta poistuu paikalta. Myöhemmin voidaan vielä palata takaisin pelastamaan loput – sille on omat prioriteettinsa. Kun vaara on ohi, aletaan pelastaa vaurioituneita kokoelman osia.

Lainaavatko pelaajat pelejä samalla tavalla kuin kirjoja?

Peliarkiston kokoelmat ovat käytössä vain paikan päällä. Pelaajat lainaavat ne tiskiltä mutta eivät vie niitä kotiin. Koska kokoelmaa käsitellään paljon, esimerkiksi levykotelot kuluvat, ja levyihin voi tulla naarmuja, joten niitä

varten on oma laatikostonsa tiskin takana. Pelaajat voivat selata koteloita hyllyillä ja sitten pyytää arkistonhoitajalta haluamansa levyn.

Onko vanhempien pelikoneiden käyttöön erityisiä sääntöjä tai käytäntöjä?

Käytännöt riippuvat paljon siitä, mistä pelikoneesta on kyse ja millaiset arkistot meillä on juuri sille koneelle. En tunne eri konsolien ja pelityyppien käytäntöjä tarkasti. Joidenkin van-

Tavoitteena on säilyttää myös pelaamisen kokemus, ei pelkkiä pelejä ja konsoleita.

hempien konsolien ja kannettavien pelilaitteiden käyttöä on rajoitettu, mutta toisaalta arkistossa monien vanhojen koneiden pelit ovat suoraan hyllyssä, josta ne voi ottaa pelattavaksi.

Miten kokoelmaa ylläpidetään ja miten pelit valitaan?

Pääarkistonhoitaja David Carter hoitaa kaiken tämän. Kokoelman kehittäminen ei ole minun vastuullani, mutta tiedän kertoa, että koska arkiston tarkoituksena on saada ihmiset käyttämään laitteita, mikään ei ole uudenveroisessa kunnossa. Usein CVGA korvaa rikkoutuneet pelit ja laitteet toimivilla. Arkistossa on varaohjaimia ja -konsoleita eniten käytetyille laitteille. Henkilökunta on myös oppinut korjaamaan rikkoutuneita laitteita. Koska tarkoituksena on säilyttää pelaamisen kokemus, mielestämme kaikki on korvattavissa.

Miten peliarkiston henkilökunta on oppinut korjaamaan pelikoneita?

Arkistonhoitajilla on jonkin verran ohjekirjoja videopelilaitteiston puhdistamiseen ja korjaamiseen. Ohjeita on myös monilla www-sivustoilla, ja

Yhdysvalloissa keräilijäpiirit ovat hyvin omistautuneet pitämään laitteistot kunnossa. Juuri heiltä suuri osa ohjeista tulee.

Miksi kokoelman eri osilla on prioriteetit? Eikö kaikki olekaan arvokasta?

Kokoelman osien priorisointi on osa yliopiston riskienhallintaa. Vaaratilanteessa ei välttämättä pystytä pelastamaan kaikkea, joten täytyy tietää, mistä aloitetaan, jos aikaa on vähän. On myös hyvä ylläpitää täydellistä listaa kokoelmien sisällöstä. Jos jotain ikävää tapahtuu, tiedämme, mitä on menetetty, mitä täytyy hankkia uudelleen ja niin edelleen.

Katastrofisuunnitelma käsittelee onnettomuustilanteisiin vastaamista, kun taas kokoelmien korvaaminen jälkikäteen on yksittäisten, eri osa-alueista vastaavien henkilöiden vastuulla. Suurissa tutkimuskirjastoissa, kuten

Vaaratilanteessa – – täytyy tietää, mistä aloitetaan, jos aikaa on vähän

meillä, kokoelmien arvoa ei voi mitata rahassa. Joitain asioita ei vain voi korvata. Katastrofisuunnittelun päämäärä on, että teemme kaikkemme, jotta välttäisiin vahingoilta.

Onko muilla yliopistoilla vastaavia peliarkistoja?

Videopeliarkistot eivät ole kovin yleisiä. Stanfordilla on yksi kokoelma ja toinen on Teksasin yliopistossa, Austinissa. Meidän arkistomme on ainoa, jossa saa pelata kaikkea. Stanford ei anna käyttäjien pelata mitään pelejä. Myös peliyhtiöillä on kokoelmia omista peleistään, mutta ne eivät ole avoinna yleisölle. Yritimme tutkia asiaa, mutta peliyhtiöiltä oli todella hankalaa saada tietoa asiasta. Tietääkseni



CVGA on täysin ainutlaatuinen sekä kokoelman kattavuuden että pelaamiskokemuksen säilyttämisen kannalta.

Tuleeko sinulle mieleen muita koelmia, jotka edes muistuttaisivat CVGA-peliarkistoa?

Löysimme paljon yhtäläisyyksiä Askwith Media Libraryn kanssa, joka arkistoi elokuvia. Lisäksi on olemassa videoarkisto Donald Hall Collection. Tutkimme, miten he ylläpitävät kokoelmiaan, ja käytämme samoja menetelmiä pelilevyjen korjaamiseen.

Mitä itse haluaisit pelastaa katastrofitilanteessa?

Haluun pelastaa asioita, jotka ovat kaikkein harvinaisimpia. Haluaisin säilyttää esimerkiksi Colecovisionin ja Commodoren konsolit, koska niiden korvaaminen on vaikeaa. Arkiston ko-

koelmaan kuuluu myös Nintendo Virtual Boy, joka on todella siisti! Emme anna ihmisten pelata sillä kovin paljon, koska se menee helposti rikki. Minun ensimmäinen videopelini oli Tetris, jota pelasin Game Boylla, joten toki haluaisin pelastaa sen myös. 🐱



- Computer & Video Game Archive:
<http://www.lib.umich.edu/computer-video-game-archive>
- Esitelmäkalvot katastrofisuunnitelmasta:
<http://www.slideshare.net/frankrd/game-over-disaster-planning-for-the-computer-and-video-game-archive-at-the-university-of-michigan>

SUOMEN PELIMUSEO

Vielä ehdit mukaan rakentamaan pelimuseota!

mesenaatti.me/suomenpelimuseo

www.suomenpelimuseo.fi



Ei näin!

SAM Coupé – imperiumin raunioilla

Kun suosittu laitteen tuotekehitys pysähtyy, muut arvelevat usein voivansa tehdä asian paremmin. Jotkut onnistuvat, useimmat eivät.

Teksti: Mikko Heinonen

Kuvat: Mikko Heinonen, Wikimedia Commons -käyttäjä TeleCarlos

Sinclair ZX Spectrum oli briteille iso juttu. Suomesakin se oli kohtalaisen suosittu mutta jäi Commodoren ja myöhemmin Spectravideon jalkoihin. Saarivaltakunnassa kuminäppäimistöjä leivottiin innokkaasti: suurin osa myydyistä viidestä miljoonasta koneesta päättyi sinne.

Sir Clive Sinclair oli kohtalainen insinööri, muttei erityisen lahjakas liikemies. Vaikka ZX-sarjan tietokoneet kävivät kaupaksi, kaikenlaiset muut projektit (kuten C5-sähköauto, pakkomielte pienien television kehittämiseen ja jokseenkin epäonnistunut Sinclair QL) veivät yrityksen talouden kuralle. Sinclair Research kaatui elektroniikkajätti Amstradin syliin. Koska Amstradilla oli myös oma tietokonemallistonsa, sitä kiinnosti lähinnä valmistaa Spectrumia niin kauan kuin se kävisi kaupaksi. Malliuudistukset olivat hyvin vaatimattomia ja niihin sisältyi etupäässä muistin ja liitäntöjen lisäämistä.

Samaan aikaan 16-bittiset kotimikrot tekivät tuloaan. Koska Spectrum oli ollut alusta asti suosituista kotimikroista teknisesti vaatimattomin, Amigan ja Atari ST:n aikakautena sen karkea grafiikka ja piipperiäännet

vaikuttivat jo todella jälkeensä. Toisaalta monella oli jo hankittuna iso valikoima Spectrum-ohjelmistoa, joten ajatus tehokkaammasta mutta Spectrumin kanssa yhteensopivasta laitteesta saattoi vaikuttaa hyvältä.

Super-Spectrum syntyy

Alan Miles ja Bruce Gordon muodostivat Miles Gordon Technologyn, joka valmisti ZX Spectrumin lisälaitteita. Heidän mielestään markkinoilla oli tilaa myös kokonaiselle tietokoneelle, joka olisi kuin Spectrum mutta tekisi kaiken paremmin. Perustaksi valittiin tuttu ja turvallinen Zilog Z80, joka kelloitettiin kuuteen megahertsiin Spectrumin 3,5:n sijasta. Muistia oli vakiona 256 kilotavua, ja 3,5-tuumainen levyasema korvasi kasetit ja Microdrivet tallennusvälineenä. Grafiikkaa terästettiin uusilla näyttötiloilla, joiden rakenne oli Spectrumia yksinkertaisempi. Äänipuolesta vastasi kuusikanavainen Philips SAA1009.

Karkea kuminäppäimistö korvattiin kallistetulla kirjoituskoneenäppäimistöllä, jonka sivuprofiilista kone sai nimensäkin. Ulkonäkö viimeisteltiin hauskoilla sinisillä tassuilla, jotka antoivat koneelle hyvin omaleimaisen ilmeen. BASIC oli varsin kehittynyt. Mikä parasta, kaikki edistyneet omi-

naisuudet olivat myös kytkettävissä pois päältä, jolloin SAM Coupé pystyi emuloimaan Spectrumia lataamalla sen ROM-tiedoston muistiinsa. Vaikka mikromarkkinat olivat hyvin täynnä, vuoden 1989 jouluna tämä olisi voinut hyvinkin olla menestysresepti valtakunnassa, jossa Spectrum tarkoitti samaa kuin C64 meillä.

Coupén kone pätkii

Ensimmäinen takaisku oli, että MGT ei saanutkaan SAM Coupéta joulumarkkinoille 1989. Suunnittelu ja valmistus viivästyivät. Osansa asiaan oli myös sillä, että RAM-muistin hinta nousi yllättäen. Näin menetettiin jo lähtökohtaisesti iso määrä asiakkaita kilpailijoille, ja pienen yrityksen pääomat jäivät kiinni myymättömään varastoon.

Koneita saatiin kuitenkin myytyä noin 8 000 kappaletta. Valitettavasti hyvin pian jouduttiin ottamaan yhteyttä täsmälleen samaan määrään asiakkaita, sillä SAMin sisälle asennettuun ROM-piiriin oli jäänyt useita virheitä. Pahin niistä esti käyttöjärjestelmän lataamisen levykkeeltä. Piirien vaihtoruljanssi tuhosi MGT:n talouden lopullisesti, ja se teki konkurssin jo keuhakuussa 1990.

Miles ja Gordon eivät kuitenkaan



olleet valmiita lopettamaan. He ostivat MGT:n varastot vasta perustetun SAM Computers Ltd:n piikkiin ja jatkoivat laitteiden valmistusta sekä myyntiä. Hintaa laskettiin, markkinoille saatiin jokusia uusia lisälaitteita ja SAMCo yritti käynnistellä ohessa myös ohjelmistotuotantoa. Esimerkiksi Eliten SAM-versio oli kuitenkin vain korpulle siirretty Spectrum-painos.

Isot kustantajat eivät SAMille juuri korviaan lotkauttaneet, sillä kohdeyleisö oli kovin rajoittunut. Pienempiä toimijoita SAM-käyttäjien rahat sentään kiinnostivat, joten muutamia kunnan pelejä saatiin: pikkufirma Fred Publishing osti oikeudet muun muassa Lemmingsin kääntämiseen SAMille.

Todellisuus saa kiinni

Vuoden 1992 puolivälissä SAM Computersin kuolonkello löi kaksitoista. Aikana, jolloin Amigakin alkoi olla vanhentunut tuoreimman PC-raudan rinnalla, ei pikkuisen piristetyllä Spectrumilla ollut todellisia mahdollisuuksia menestyä. West Coast Computers -niminen yritys osti loppuvaraston ja tekaisi ”uuden” SAM Elite -mallin tuplaamalla muistin 512 kilotavuun. Se ei aiheuttanut ostoryntäystä.

SAM Coupé ei lopulta ollut ihan niin Super-Spectrum kuin voisi luulla. Suoritin oli kyllä hieman nopeampi, mutta grafiikan toteutustavasta johtuen se joutui väistämään ASIC-piiriä muistia käyttäessään. Käytännön laskentateho ei paljonkaan esikuvasta eronnut. Kotikutoisuuden vuoksi siellä ja täällä

oli myös ominaisuuksia, jotka kielivät viimeistelyn puutteesta: resetoimalla kone sopivassa kohtaa oli muun muassa mahdollista tuhota koko levykkeen sisältö. Kahta peliohjainta yhtä aikaa käytettäessä ne häiritsivät toisiaan. TV-modulaattori oli asennettu virtalähteen sisään, mikä ei ainakaan vähentänyt kuvan häiriöiden määrää.

Suurin ongelma oli kuitenkin kaiketi se, että taaksepäin yhteensopivuuden merkitystä oli yliarvioitu. Spectrum oli jo SAMin julkaisun aikaan vanha kone, josta monet olivat valmiit päättämään irti modernimman tarjonnan edessä. Parannukset olivat niin lieviä, että laitteesta oli lopulta iloa lähinnä kovimmille faneille ja ohjelmoinnin harrastajille. Uuden Amstradin valmistaman Spectrumin kun sai edelleen kaupasta, jos etsi pelkkää pelilaitetta. Myyntitavoitteen saavuttaminen olisi ollut hankalaa jo vuoden 1989 joulumarkkinoilla, ja kun ne lipsahtivat käsistä, peli oli pelattu.

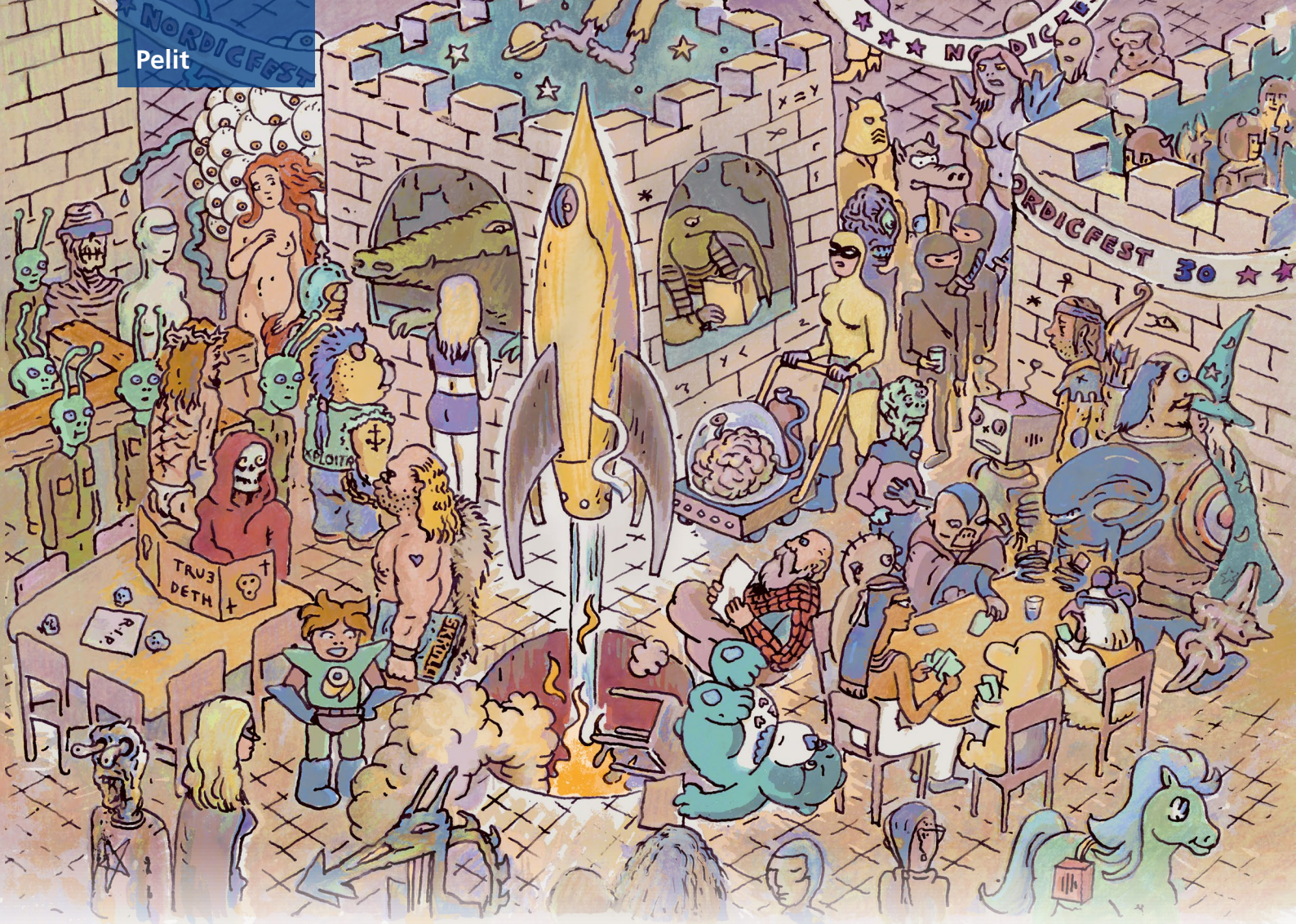
Samanlainen tarina on nähty monen muunkin laitteen haaskalla: esimerkiksi yksikään Commodoren konkurssin jälkeinen C64- tai Amiga-uusintaversio ei singahtanut maailmanmenestykseen. Ne jättivät jälkeensä samanlaisia vanoja konkurssiin menneitä yrityksiä ja sekalaisia oikeuksien omistajia. Kirous hälväni vasta 2000-luvulla, kun kentälle ilmestyi piensarjoihin erikoistuneita pikkufirmoja, jotka eivät tavoitelleet kuuta taivaalta.

Sisäpiirin lempilapsi

Kun West Coast Computers laittoi luo- kut kiinni vuoden 1994 tienoilla, SAM Coupén maailmanvalloitus oli virallisesti päättynyt. Arvioiden mukaan noin 12 000 hassua sinitassuista brittimikroa oli päättynyt ostajille. Moni voisi arvella, että homma päättyi tähän.

Tietokoneharrastus on kuitenkin siitä jännittävä, että mitä erikoisempi tai marginaalisempi laite on kyseessä, sitä varmemmin se löytää oman harrastajapiirinsä. SAM Coupén kohdalla tämä piti paikkansa erityisen hyvin. Kun virallista ohjelmistoa ei käytännössä ilmestynyt, ottivat yksittäiset harrastajat asiakseen tukea konetta omilla tuotoksillaan. Isossa-Britanniassa ilmestyi useampikin fanzine ja diskettilehti, ja saatavana oli myös paljon erilaisia lisälaitteita aina IDE-kiintolevyohjaimen asti.

Vähäisestä valmistusmäärästään johtuen SAM on nykyisin hintava aparaatti. Verkkohuutokaupoissa toimivat yksilöt maksavat jo nyt satoja euroja, laatikon kanssa tonnin. Ei siis ihme, että harrastus ei juuri houkuttele uusia ystäviä mutta entiset tuntevat jatkavan toimintaa ahkerasti. SAM Revival -diskettilehti ilmestyy edelleen, ja rautapuolella saatavana on mm. kortinlukija, verkkokortti sekä adapteri C64:n SID-äänipiirille. Peljä sekä skenedemoja tehtaillaan myös yhä. Kenties aktiivisin toimija Quazar on pyrittänyt omaa SAM-verkko-kauppaansa jo 21 vuotta osoitteessa www.samcoupe.com. 🐘



Satunnainen samooja

Kolme vuosikymmentä suomalaista roolipelaamista, osa 2

Suomalaisen roolipelaamisen veteraani, Peliluolan suurmestari Nordic jatkaa historiankirjojen tutkimista. Vuosituhannen jälkeen edessä odottivat saatanalliset säkeet, harrastuksen hiipuminen ja uusi nousu yllättävin keinoin.

Teksti: Nordic the Incurable Kuvat: Mikko Torvinen, Nordic the Incurable

Tämä artikkeli on jatkoa Skrollin numerossa 2015.2 julkaistulle Satunnainen samooja -jutulle.

Vuosituhatteen vaihteen lähestyessä Claymore-roolipilehti hiipui tekijöiden kii-reiden ja suppean le-vikin vuoksi. Roolipelaamisen suosio oli tuohon aikaan vielä kohtalaisen korkealla, mutta jostain syystä suomalaiset roolipelaajat eivät koskaan oikein innostuneet alan lehdistä. Edes ulkomaiset, Dragonin kaltaiset varsin laadukkaat lehdet eivät menneet pelaajille kaupaksi.

Kotimainen Magus jaksoi sitkeästi yrittää. Sillä oli vuositilaajiaakin jonkin-moinen määrä, mutta ei siitäkään mitään varsinaista hittiä tullut. Toki neljä numeroa vuodessa oli myös rauhallinen ilmestymistähti. Edellinen numero ehdittiin lukea muutamaan kertaan läpi uutta odotellessa.

Lehden tekemisen ilon ohella hom-masta koitui muutakin hyvää. Claymoren ja muutamien aiempien painotuotteiden valmistusprosesseissa olin havainnut tamperelaisen, jo edes-menneen Tehokopioinnin erinomaisen edulliseksi paikaksi. Niinpä aloin teettää siellä pieniä A6-kokoisia pelejä sadan kappaleen painoksina, osittain tekemisen ilosta ja osin siksi, että painattaminen oli niin halpaa. Ensimmäinen peli taisi olla Kalmo, sittemmin lähes legendaarisen maineen saanut zombieroolipeli, jossa pelaajahahmot ovat zombeja. Kalmo meni hyvin kaupaksi, ja siitä otettiin ainakin viisi, aina hiukan uudistettua painosta.

Kalmon hyvän vastaanoton siivit-tämänä tein liveroolipeli Elokalmon. Peleissäni alkoi vilistä yhä oudompia aiheita. Alkeiseliöroolipeli Ameba,

maalaisroolipeli Inkkunen, Ankkunen, Ynkkyinen, Hakomäki sekä lelu-roolipeli Leluelo ilmestyivät muuta-man vuoden sisään. Vaikka pelien aihe oli outo tai humoristinen, ne olivat yksinkertaisuudestaan huolimatta pel-lattavia. Erityisesti Inkkunen jne. -ropen parista muistan muutamia todella hauskoja ja vauhdikkaita seikkailuja – unohtamatta tietenkään Kalmoa, jota pelasimme useaan otteeseen.

Jo mainittujen lisäksi tunsin tarvetta tehdä soolopelin nimeltä File X, Warhammer-maailmaa parodioivan figupeli Fartslammerin, absurdin sarjakuvan mestaria Jukka Tilsaa kunnioittavan Tilsaworldin sekä mainion, silloisesta D20-hypestä nimensä saaneen Väinö D20:n. Näistä erityisesti Väinö kohosi suosioon ja on ollut vielä vuosien jälkeenkin kysytty peli.



Suomessa julkaistiin useaa harrastajapohjaista roolipelilehteä, mutta hiteiksi ne eivät nousseet.

Eikä ihme: se on erinomaisen hauska ja järjetön roolipeli, jossa ei tarvita lainkaan pelinjohtajaa. Kalevalaiseen maailmaan sijoittuvassa Väinössä pelihahmoksi voi valikoitua länsituuli, saunantakunen tai vesikirppu.

Jo näihin aikoihin minulla oli usein uutta peliä suunnitellani tavoitteena tehdä peli, jonka kaltaista ei vielä ollut olemassa. Taisin onnistuakin aika

hyvin, koska en vielä kukaan ole kuullut, että kukaan muu olisi tehnyt roolipeliä ameboista, leluista tai maalaistolvanoista.

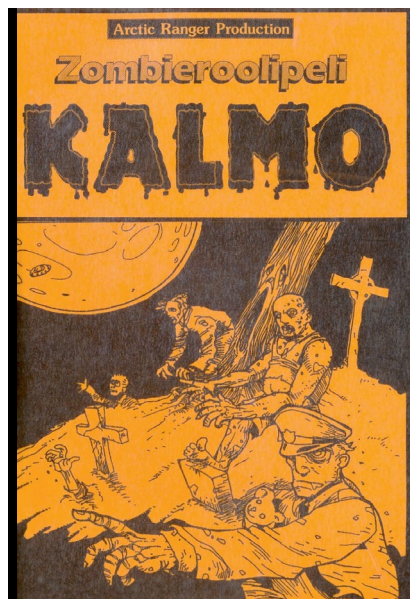
Pahuuden teillä

1990-luvulla oli liikkeellä ulkomaan huhuihin perustuvia lehtijuttuja, joissa kauhisteltiin roolipelien mahdol-

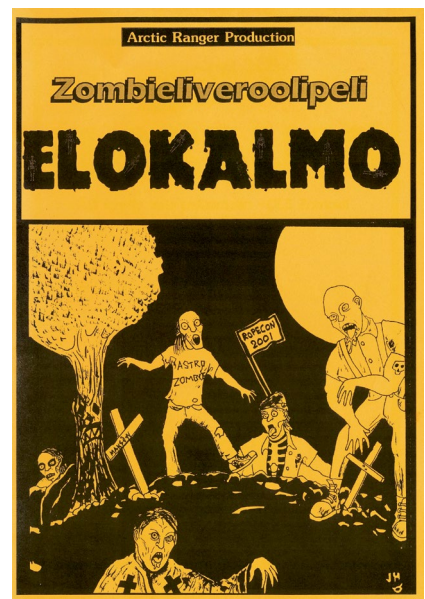
lista saatanallisuutta. Olipa täällä kotomaassakin joitakin yksilöitä, jotka vailla suurempaa tietoa saarnasivat ropeilun vaaroista. Eräässä tapauksessa huolestunut kirjoittaja kauhisteli sitä, kuinka roolipelissä ”pelinjohtaja pakotti pelaajat syömään koiransa”. Kyseessä oli ropeseikkailun kuviteltu tilanne, jossa seikkailijaseurue oli



Suomalaiset roolipelaajat eivät innostuneet roolipelilehdistä.



Zombieroolipelit Kalmo ja Elokalmo aloittivat pienten lehtiökokoisten roolipelien sarjan. Pikkusarjoina tehtävissä seikkailuissa oli tilaa myös oudoille ideoille.





Sattuneesta syystä Roolipelaajan saatananpalvonta-opasta myytiin vain tiskin alta.

ansassa nälkäkuoleman partaalla. Tälöin pelihahmot joutuivat turvautumaan koirapaistiin selvitätseen hengissä. Mielikuvitus oli ilmeisesti tuntematon käsite monille pelien vastustajille.

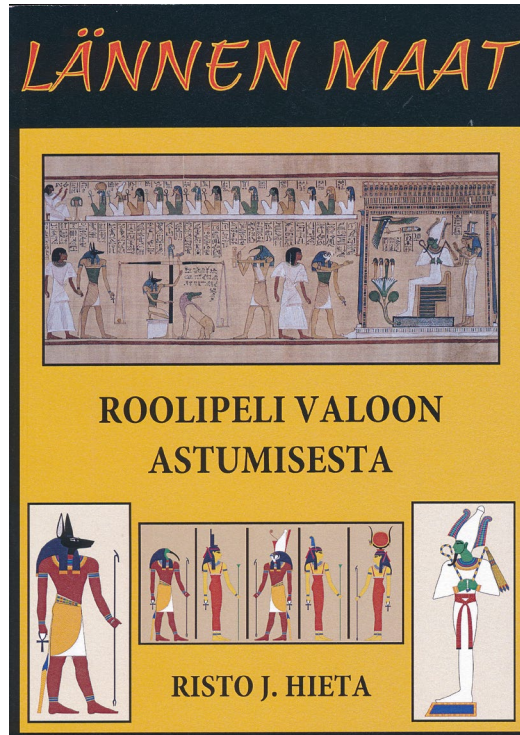
Osittain noiden tapausten innoittamana väsäsin jossain välissä humoristisen pikku vihkosen nimeltä Jokaisen roolipelaajan saatananpalvontaopas. Fantasiapeli Lauri Tudeer, joka mielihyvin otti pikkupelini myyntiin, kuitenkin katsoi, että ärsytyskynnyksen mahdollisesti ylittävä opas ei kuulunut Fantsun tiskille. Niinpä sitä myytiin tiskin alta. Hyvin sekin meni kaupaksi, sillä minullakaan ei ole jäljellä enää kuin pari kappaletta.

Pikkupelien teko oli mukavaa, vaikka eihän siitä rahallisesti hyötynyt. Niiden tekeminen kuitenkin jäi pitkäksi aikaa, kun Tehokopiointi lopetti ja metetin edullisen kopiopaikkani.

Kunnon kampanjat

Peliporukassamme pelattiin 90-luvun lopulla monia hyviä kampanjoita. Pidemmistä muistuvat mieleeni hyvät seikkailut muun muassa Vampiren, Wraithin, Earthdawnin ja Fading Sunsin parissa.

Käytimme myös kuukausikaupalla aikaa Deadlandsin preerioilla. Se oli hyvin miellyttävä tuttavuus. Asi-



Yksi pelaaja ja monta pelinjohtajaa? Muinaisen Egyptin mytologia innoitti Lännen maat -roolipeliä, josta tulikin erikoinen luomus.

aan vaikuttivat niin mielenkiintoinen maailma, toimivat säännöt kuin pelinjohtajamme Hansin syvälinen paneutuminen. Hänen ansioihinsa lukeutui muun muassa upeat, vanhaan valokuvatyyliin taiteillut kuvat hahmoistamme.

Deadlandsissa tapahtumien taustana on villi länsi höystettynä oudolla tekniikalla, henkimaailmalla ja ylimaailmallisilla otuksilla. Hahmomme vaikuttivat pienessä lännen kaupungissa, jonka "rakentamiseen" kaikki pelaajat osallistuivat. Pelihahmojen elämässä oli paljon muutakin kuin vain rosvojen takaa-ajoa ja taistelua. Muutamilla oli aktiivista sosiaalista ja perhe-elämää, ja eräskin ikimuistoinen pelikerta keskittyi siihen, kun seriffi ja lakimies lähtivät junalla isoon kaupunkiin ostamaan puolisoitensa ohjeiden mukaisia astiastoja. Sirpaleista huolimatta kaikki päättyi lopulta hyvin.

Fading Suns oli hiukan erilainen scifiroolipeli. Siinä avaruus oli laajalti asutettu ja alukset kulkivat planeettojen välillä. Maailma oli vanhanaikaisen feodaalinen luokkayhteiskunta, jossa valta todella oli niillä, joilla se oli. Mahtisuvut kamppailivat alueista, rahasta ja vallasta, ja pelihahmot koettivat valita viisaasti oikean puolen. Maailma sisälsi paljon mielenkiintois-

ta nähtävää, koettavaa ja kokeiltavaa, ja se jäi mieleen mainiona vaihtoehtona perus-Star Wars-tyyliselle avaruusräiskinnälle.

Tarinankerron-tapelit, kuten niitä nimitettiin, olivat kova sana 90-luvulla. Meikin pelasimme niistä ainakin Wraithia ja useaan otteeseen Vampirea. White Wolf -yhtiön tuottamassa pelisarjassa vaikuttivat mainittujen lisäksi sellaiset pelit kuin Werewolf, Mage, Changeling ja Trinity. Nämä olivat tuolloin aiheiltaan uusia ja omaperäisiä, ja kun niitä vielä markkinoitiin hy-

vin, saivat ne aikaan meillä Suomessa-kin pienimuotoisen World of Darkness-hypetyksen.

Tarinankerronta tai siihen kannustaminen eivät säännöissä tulleet mitenkään vahvasti esille. Myöskään pelien säännöt eivät olleet erityisen kevyitä eivätkä joustavia. Ne oli kuitenkin suunniteltu hyvin, ja pelit tarjosivat mahdollisuuden hauskaan pelaamiseen, mutta ihmeellisen harva pelaaja huomasi koskaan kyseenalaistaa niiden "tarinankerrontaa". Itse en sitä ainakaan koskaan havainnut yhtään enempiä kuin muissakaan peleissä, ja oman porukkamme tyyli on ollut aina melkoisen vapaa ja ennakkoluuloton.

Olen aina – sekä pelaajana että pelinjohtajana – arvostanut sitä, että hahmoilla on yrittämisen vapaus. Säännöt tai seikkailun raamit eivät saa kahlita hahmojen kokeilumahdollisuuksia. Pelinjohtajalla täytyy olla sen verran sävellyskykyä, että hän osaa suhtautua pelihahmojen erilaisiin temppeihin oikein.

Muutaman kerran olen törmännyt tilanteeseen, jossa hahmo yrittää tehdä jotain yllättävää ja pelinjohtaja vain ilmoittaa ilman perusteluita: "Et voi tehdä sitä." Aaargh! Tuollaista minun on vaikea kestää. Vaikka suon mielelläni pj:lle kaiken vallan, täytyy hänen osata

käyttää sitä viisaasti. En osaa kovasti arvostaa pelinjohtajaa, jonka täytyy asetella hahmojen ympärille näkymätömiä seinä.

Vanhat pelaajat

Vuosituuhannen vaihduttua roolipeli-harrastuksen hiipuminen näkyi yhä selvemmin. Suurelta osin se johtui uusien pelaajien puutteesta. Nuoriso oli tottunut digitaaliseen pelaamiseen, ja erilaiset pelikoneet tarjosivat jo varsin kiinnostavia pelimaailmoja. Oli vaikea keksiä mitään syytä, miksi nuoren kannattaisi tarttua 200-sivuiseen kirjaan ja ryhtyä kuvittelemaan fantasia-maailman tapahtumia, kun ne olivat jo hänen edessään ruudulla elokuvamaisen näyttävänä toimintana.

Valitettavasti sama pätee edelleen. Vanhat roolipelaajat jatkoivat pelailua, mutta vapaa opiskelijaelämä vaihtui pikkuhiljaa työhön ja perhe-elämään. Nämä vaikuttivat osaltaan monen peliryhmän yhä harvemmiksi käyviin ropeiltoihin.

Itselläni on käynyt niin hyvä onni, että pelaamiseni on jatkunut melkoisen katkeamattomana 1980-luvun lopulta saakka. Pelaamme kerran viikossa ja vain muutaman kerran vuodessa saatamme pitää parin kolmen viikon tauon joului-, kesä- tai muiden lomien vuoksi. Vaihtuvuutta peliryhmämme kokoonpanossa on luonnollisesti ollut, mutta sekin on ollut perin hillittyä.

Nykyisestä joukosta minun lisäksi Skide on pelannut alusta asti ja ollut mukana koko ajan YK-komennusta lukuun ottamatta. Hanski on ollut vakiohemmo yli 20 vuotta ja toiminut usein myös pelinjohtajana. Vilekin lieinee hengailnut mukana lähes kymmenen vuotta, ja Kallella on kokemusta porukassamme noin kolmen vuoden verran.

Aiemmista sankareista Archie kuului alkuaikojen vakiokalustoomme monen vuoden ajan mutta poistui sitten elämään oikeaa elämää. Japi oli mukana joukossamme melkein kaksikymmentä vuotta. Elävä legenda, Pete, oli alkuaikoina mukana monissa taistelullisissa kampanjoissa, etenkin Cyberpunkissa ja Warzonest. Tappioita syntyi niin omille kuin vastustajillekin, onneksi yleensä enemmän jäljempänä mainituille. Mano, Antti, Tommi, Paula, Kimbe ja monet muut ovat vierailleet peliporukassamme vuosien

aikana. Ja mikä ihmeellistä, kaikkien kanssa on ollut upeaa pelata!

Pelitekoja

2010-luvulle tultaessa olin pitänyt pitkähköä taukoa omista julkaisuistani. Ideoita kyllä oli, ja omaan käyttöön tein niin rooli- kuin lauta- ja korttipeljäkin. Jossakin vaiheessa suunnittelin muinaisen Egyptin mytologiaan liittyvää peliä. Idea jäi hautumaan, ja kun kävin museokeskus Vapriikin Egypti-aiheisessa näyttelyssä, innostuin työstämään peliä kunnolla.

Pelimalliksi muotoutui roolipeli, koska lautapelejä aiheesta oli jo useita. Muutamien yleisroolipelien lisäosista löytyi Egypti-aiheita, mutta täysimittaista Egypti-ropea en löytänyt. Halusin tehdä jotakin erilaista ja uutta. Tuomasin, että maailmassa tuskin on kovin monta ropea, jossa on yksi pelaaja ja monta pelinjohtajaa. Sellaisen voisin siis tehdä.

Egyptin rikas mytologia tarjosi paljon mahdollisuuksia. Kun itsestään ajatukset johtivat siihen, että pelinjohtajat ovat tietysti muinaisia jumalia. Pelaajalle jäi näin kuolleen egyptiläisen osa. Pelaajan täytyy kuljettaa vainajaa manalassa jumalten asettamien haasteiden ohi ja lopulta selviytyä heidän suorittamastaan ”kuulustelusta” autuuteen päästäkseen. Lännen maat -pelistä tulikin hyvän taustatyön ansiosta erinomaisen toimiva.

Kustantajaksi löytyi kotimaista pelintekoa edustava Myrrysmiehet, ja peli julkaistiin Ropeconissa 2013. Se on seurapelimäisen helppo, mutta siinä on myös roolipelimäistä vapautta ja päätöksentekoa. Jopa Vapriikki kelpuutti Lännen maat myymälänsä valikoimiin.

Huomasinpa sitten opeturopekipailun netissä. Aikaa oli muutama päivä. Kun olin kerran jo uppoutunut Egyptiin, niin enpä siinä sen enempää miettinyt vaan tarjouduin tekemään aiheesta opetuspelin. Pääsin mukaan yhtenä valituista ja väänsin varsin toimivan ja selkeän Valon maa -pelin Otavan opiston käyttöön. Pelin voi ladata ilmaiseksi heidän sivuiltaan.

Seuraavana vuonna kehitin Marsin agentit -huumoripelin. Siinä oli erikoista parviällyn käyttö, eli pelihahmot ovat yhdessä fiksumpia kuin yksinään. Pelistä tuli odotusten mukainen, mutta siitä olisi tullut vielä parempi, jos olisin

osannut miettiä siihen mukaan Sami Kopenen myöhemmin ehdottamia parannuksia.

Tänä vuonna Ropeconissa julkaistiin jälleen Myrriksen kustantamana humoristinen, joskin teemaltaan kam-mottava seurapeli Tupilak. Sen erikoisuuksia ovat sekä aihe (kuolema Grönlannissa) että taulukoihin perustuva pelaaminen. Se sopii mille tahansa pelaajamäärälle eikä vaadi minkäänlaista pelinjohtamista eikä muuta kontrollia. Peli taidettiin myydä loppuun jo Ropeconissa, mutta siitä tulee toivottavasti pian uusi painos.

Ropenäkymät

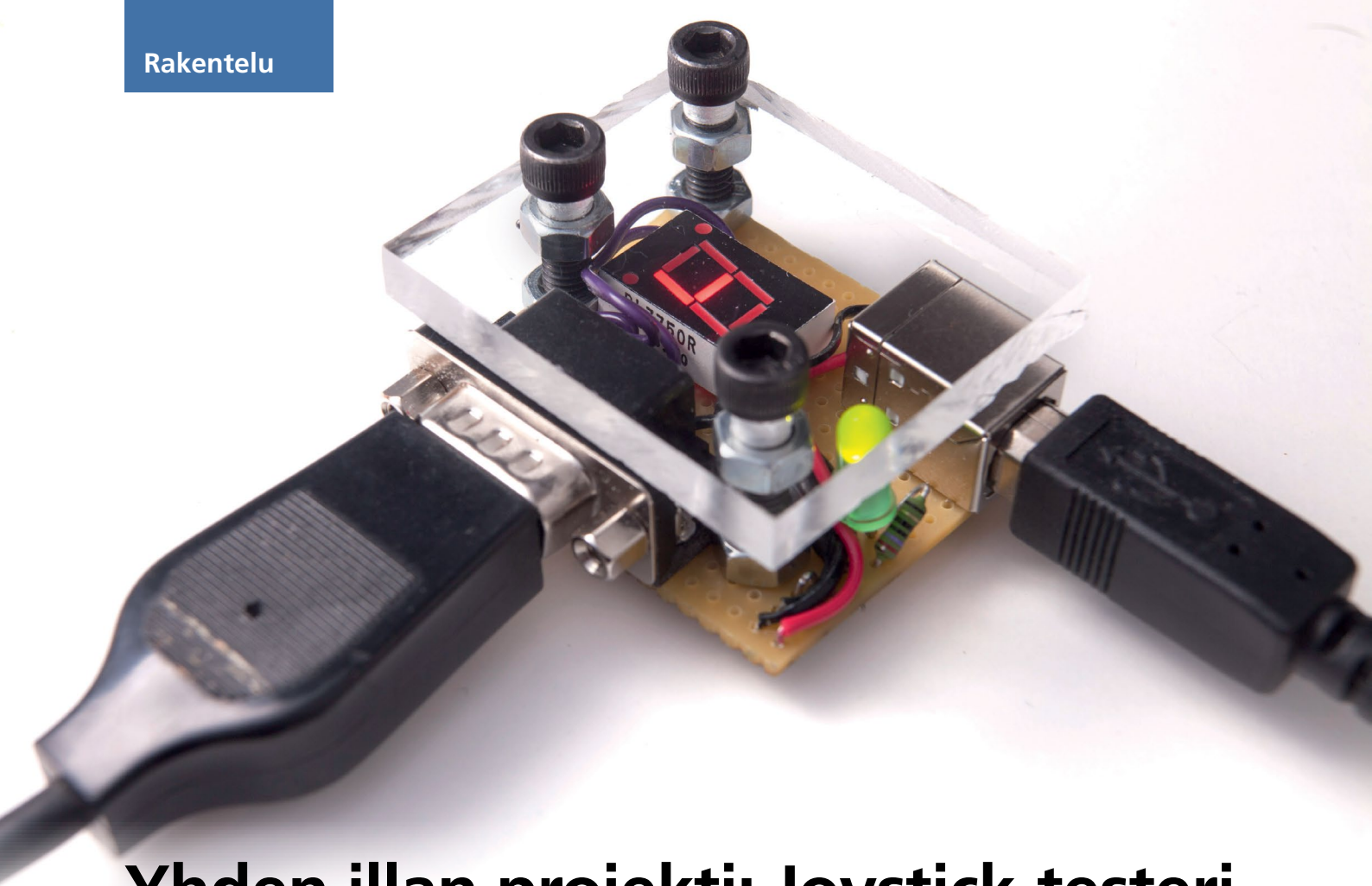
Roolipelien suosio on sekä maailmalla että meillä kovin alhainen verrattuna 15–20 vuoden takaiseen aikaan. Kyllä pelejä yhäkin tehdään, myydään ja pelataan, mutta pelikaupan ropehylly on kovin säälittävä vanhoihin hywiin aikoihin verrattuna. Sieltä löytyy lähinnä Dungeons ja Dragonsia, joiden kaverina on Pathfinderit, Praedorit ja muutama muu peli. Loppuja myydään yksi tai kaksi kappaletta vuodessa.

Lautapelit ovat puolestaan nousseet suosion harjalle, eikä se ole lainkaan huono asia. Niistä löytyy nykyään niin mahtavan upeita systeemejä, näyttäviä pelilautoja ja hienoja maailmoja, että parhaimmillaan ne tarjoavat lähes roolipelimäisiä elämyksiä. Omassakin peliporukassamme yhä useampi pelilta kuluu lauta- tai korttipelien parissa. Kaikilla meillä olisi kyllä halua pelata ropejakin, mutta kenelläkään ei taida olla aikaa tai intoa ryhtyä pelinjohtajaksi.

Vuosia sitten tekemäni viikinkiaiheinen kampanjakin odottelee edelleen pelauttamista. Ehkä sitten joskus... 🐉

Kuka Nordic?

Suomalaisen roolipelimaailman grand old man Nordic the Incurable on tehnyt ja julkaissut vuosikymmenten mittaan kymmeniä roolipelejä, pelannut läpi valtavan määrän muiden luomuksia ja kirjoittanut roolipeleistä viiteen eri lehteen. Tietokonepelaajat muistavat Nordicin Mikrobitti-lehdessä ilmentyneestä Peliluola-palstasta, jonka kautta Nordic loi Suomeen kukoistavan pohjan sekä tietokone- että perinteisille roolipelaajille. Nordic työskentelee yhä pelialalla, tekee edelleen roolipelejä ja pelaa viikoittain.



Yhden illan projekti: Joystick-testeri

Itse tehdyt laitteet on tärkeää testata rakentamisen ja käyttöönoton välissä. Jopa joystick voi väärin rakennettuna aiheuttaa koneelle vahinkoa. Tämän jutun ohjeita joystick-testerin rakentamiseen voi soveltaa melkein minkä tahansa nappeja kytkevän ohjaimen testaamiseen.

Teksti: Ville Ranki, Risto Mäki-Petäys, Annika Piironen Kuvat: Risto Mäki-Petäys

Olen rakentanut viime vuosina useamman Atari-liitännäisen joystickin, koska suosikkiohjaimeni TAC-2 on turhan pieni aikuisen kouraan. Tikkuja rakennellessa voi kuitenkin aina mokata ja kytkeä johtoja väärin. Tällöin parhaassa tapauksessa tikku ei toimi ja huonoimmassa se voi aiheuttaa vahinkoa tietokoneelle. Esimerkiksi Commodore 64:n joystick-liitännän nastassa 7 on 5 V:n jännite, jonka suoraan maahan kytkevä muun muassa näppäimistöä ja joystickia lukeva CIA-piiri hajoaa. Tämän vuoksi tikun toimivuus olisi hyvä voida testata ennen tietokoneeseen kytkemistä.

Saatuani valmiiksi uusimman tikku-ni minulla ei ollut turvallista tapaa sen testaamiseen. Niinpä päätin rakentaa joystick-testilaitteen, joka valmistui parin tunnin puuhastelun jälkeen.

Tavoite

Rakennetaan joystick-testeri. Testeri sytyttää ledejä palamaan sen mukaan, mitä kytkentöjä kahvan sisällä sulkeu-

tuu sitä testattaessa. Laitteen käyttäjä tarkkailee, syttyvätkö testiliikkeitä vastaavat ledit.

Osat ja työkalut

- verolevyä (viivarasterilla)
- kolvi ja tinaa
- tinaimuri tai tinaimusukka (epäonnistuneitten juotosten korjaamiseen)
- ledejä tai 7-segmenttinäyttö (käytetään suuntien ilmaisemiseen)
- vastuksia
- uros-DB9-liitin
- virransyöttö (USB-johto tai -liitin, paristokotelo tms.)
- yleismittari
- ehjä joystick (testaamiseen)

Osien hinnaksi tulee muutamia euroja. Jos työkaluja ei omista entuudestaan, ne voi ostaa yhteensä muutamalla kymppillä. Rakensin oman testerini Tampereen Hacklabilla, jossa kaikki osat ja työkalut löytyivät elektroniikkatyöpisteen tuoilta nousematta.

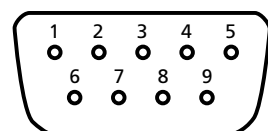
Nastat ja kytkennät

Atari-joystickin toiminta perustuu

sihen, että se yhdistää kytkimillä eri signaalinastoja maahan (esimerkiksi ylös, alas, tulitus). Atari-joystickit (*autofirea* lukuun ottamatta) eivät tarvitse käyttöjännitettä tai älyä, kuten esimerkiksi USB-liitännäiset joystickit. Tikku kytketään 9-nastaisella D-liittimellä (DB9, kuva 2) tietokoneeseen. Tikussa on naaras- ja koneessa urosliitin. Nastojen numerot on merkitty usein fyysiseen liittimeen pienillä numeroilla. Testerin kannalta Atari-joystick on viisi kytkintä.

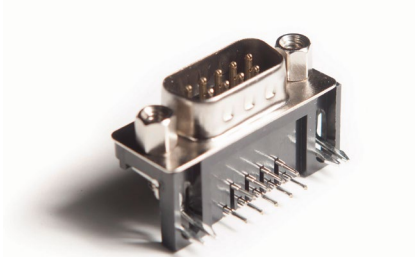
Ledit palamaan

Ledien sytyttäminen vaatii sähköä, jota saa kätevästi muun muassa USB-liittimistä, joissa on viiden voltin jännite.

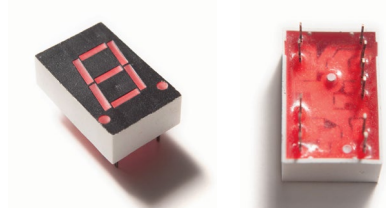
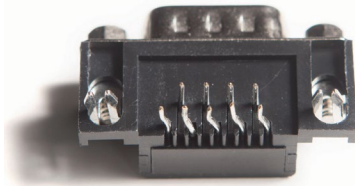


Kuva 1. D-liittimen kaaviokuva. Nastat järjestyksessä:

1. Ylös. 2. Alas. 3. Vasemmalle. 4. Oikealle.
5. Potentiometri pysty. 6. Tulitus. 7. +5 V.
8. Maa. 9. Potentiometri vaaka.



Kuva 2. Piirilevyllä juotettava D-liitin. Viereisessä kuvassa sama liitin, jonka lomittaiset jalat on taivutettu sopimaan verolevyllle.



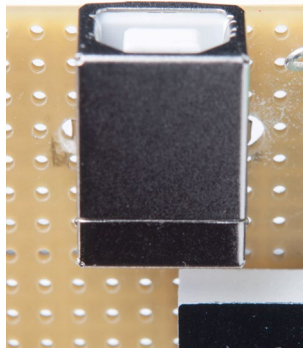
Kuva 3. 7-segmenttinäyttö kuvapuolelta sekä alhaalta.

USB-kaapeleissa käytetään standardoitua värimerkintää, joten johdon kuoren alta löytyy punainen (plus) ja musta (miinus tai maa) johto.

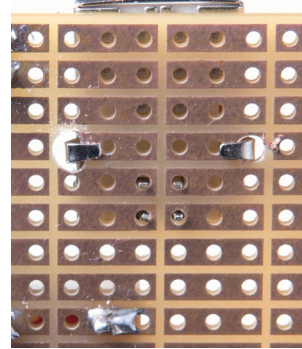
Ledi on diodi, jonka voit kuvitella johdonpätkäksi, jonka läpi virta pääsee vain yhteen suuntaan. Jos kytket ledin plus- ja miinusnavan väliin virtalähteeseen toisinpäin, mitään ei tapahdu, koska virta ei kulje. Päinvastoin kytkemällä saat aikaan oikosulun, jolloin ledi vilkahtaa viimeisen kerran. Tätä ei kannata kokeilla tarkoituksella, sillä virtalähde ei välttämättä pidä oikosulusta.

Ledi tarvitsee kaverikseen etuvastuksen, joka hillitsee sen läpi kulkevaa virtaa. Viiden voltin jännitteellä sopiva etuvastuksen resistanssi on ledistä riippuen suurin piirtein 330Ω . Suuremmalla resistanssilla ledi palaa himmeämmin ja pienemmällä kirkkaammin, ääritapauksessa rikkoutuen.

Diodin suunnalla on väliä. Ledeissä pidempi jalka osoittaa plussan ja lyhyempi miinuksen. Vastuksen suunnalla

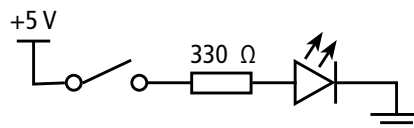


Kuva 4. USB-runkoliitin paikallaan. Verolevyllä joutuu tekemään sopivat reiät.



Kuva 5. USB-kaapeli kuorittuna.

ei ole väliä. Lisäksi ei ole väliä, kummalla puolen lediä etuvastus on, sillä se jarruttaa virran kulkua molemmissa tapauksissa. Piirissä olevan kytkimen paikalla ei ole myöskään merkitystä. Kuvassa 1 on esimerkkikytkentä kytkimellä syttyvälle ledille.



Kuva 6. Esimerkkikytkentä kytkimellä syttyvälle ledille.

Vaihe 0

Hahmottele verolevyn ja osien kanssa, miten saat mahtumaan verolevyllä osat sekä niiden ympärille tilaa hyppylangoille. Jätä varmuuden vuoksi verolevyn linjojen leikkauskohtiin tilaa viillon tekemiseen.

Mieti, mihin kohtaan verolevyä näyttö (tai ledit) kannattaa laittaa niin, että tarvitsee tehdä mahdollisimman vähän juotoksia. Täplästeriverolevyllä voit yhdistää vierekkäisiä reikiä yhteen tinasillalla, mutta pidemmät vedot kannattaa tehdä hyppylankoina.

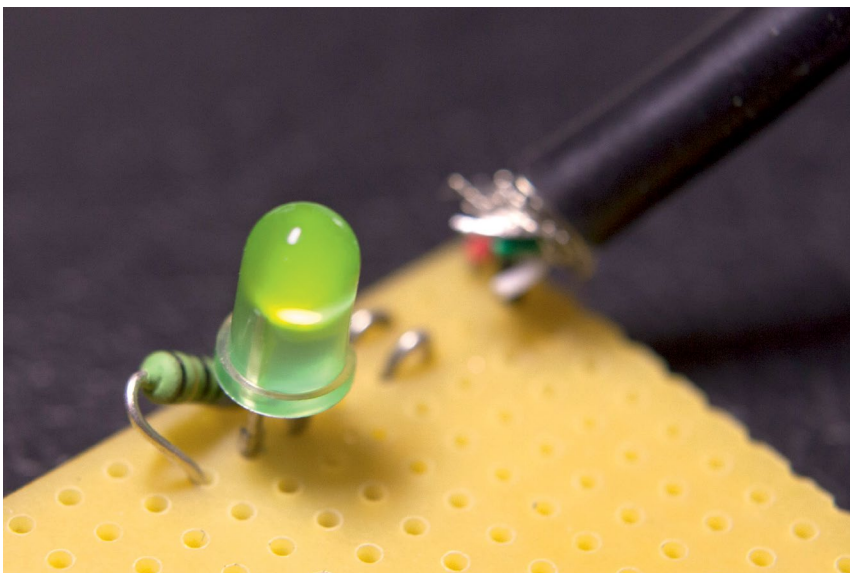
On mahdollista suunnitella kytkentä lennosta, mutta jos olet epävarma, niin piirrä kytkentäsi ensin vaikkapa ruutupaperille.

Vaihe 1

Kytke virransyötön nastat (+ ja -) verolevyn reunaan.

Jutun esimerkkilaitteessa piirilevyllä juotettiin USB-runkoliitin, johon voi kytkeä sopivan USB-johdon.

Verolevyyn voi juottaa myös piikki-riiman pätjän, jolla voi syöttää sähköä jostain muusta virtalähteestä ilman USB-isäntälaitetta tai -keskitintä.



Kuva 7. Ensimmäinen rajapyykki saavutettu: ledi palaa.

Vaihe 2

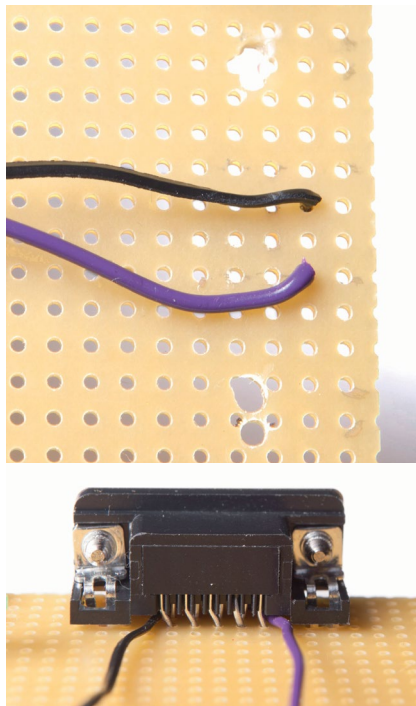
Tee harjoitusmielessä virtavalvovalle. Tarvitset vastuksen ja ledin. Tarkista yleismittarilla ennen kytkemistä, että virransyötön nastojen välille ei ole syntynyt siltoja (oikosulkuja). Tarkista, että valo syttyy, kun kytket virtalähteen kiinni.

Ohjaimen liittimen kytkeminen

Projektin mekaanisesti haastavin osa on D-liitin. Löysin Hacklabista vain liittimiä, joissa on 90 asteen kulma piirilevylle kiinnittämistä varten. Valittavasti ylä- ja alarivien nastat ovat hieman lomittain, eikä liitin sovi suoraan verolevylle. Ratkaisin ongelman tyyliä taivuttamalla yläriviltä tulevat nastat kärkipihdeillä sivulle niin, että ne sopivat verolevyn reikiin (kuva 2).

Jos tämä ei onnistu, voit pistää DB9-liittimen johdonpätkien perään, jolloin joudut kiinnittämään sen koteloon kunnolla. Toisaalta tällöin tarvittavien hyppykankojen määrä vähennee. Huomaa, että kaikkia yhdeksää nastaa ei tarvita – suunnat, tulitus ja maa riittävät. Päätä, miten kiinnität liittimen.

Piirilevylle kiinnitettävää liittintä käytettäessä huomioi, että etummaisat nastat jäävät liittimen alle. Kannattaa siis ensin kytkeä niiden johdot piirile-



Kuva 8. Verolevylle D-liittintä varten porattavat reiät. Liittimen alle jäävät johdot on vedetty valmiiksi ennen liittimen asennusta.

vyllä ja vetää johdot liittimen alta – sitten vasta kiinnittää liitin (kuva 8).

Vaihe 3

Kytke D-liittimen nastat verolevylle. Tarkista, ettei nastojen välille ole syntynyt siltauksia.

Näytön rakentaminen

Aluksi oli tarkoitus tehdä testerin näyttö viidestä ledistä, jotka syttyvät tikkoa vääntäessä tai nappia painaessa. Päädyin kuitenkin käyttämään 7-segmenttinäyttöä (kuva 2). Näyttöjä on kahdenlaisia: yhteisanodinäytössä (common anode) on yhteinen plusnapa ja segmenttejä vastaavat miinusnavat ja yhteiskatodinäytössä (common cathode) yhteinen miinus ja segmentteille plus.

Jos käytät irrallisia ledejä, niin voit tehdä näytöstä vapaammin haluamasi kaltaisen, mutta joudut vetämään jokaiselle ledille erikseen plussan ja miinuksen.

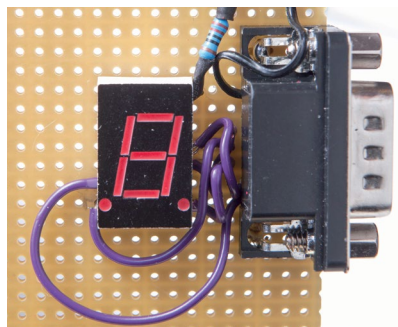
Tässä ohjeessa kaikilla ledeillä on yhteinen etuvastus, jolloin useamman ledin palaessa samanaikaisesti kirkkaus himmenee. Ahkerampi rakentaja voi halutessaan kytkeä kullekin ledille oman etuvastuksen. Tämä ratkaisu on käytännöllisempi myös, jos käytetään erityisen tehokkaita ledejä.

Anodi: + tai positiivinen napa.
Katodi: –, maa tai negatiivinen napa.
Tässä jutussa käytetään tekstin lomassa plus- ja miinustermejä luettavuuden vuoksi.

Vaihe 4a (jos käytät näyttöä)

Selvitä, miten näyttösi toimii, ja suunnittele sen kytkentä.

Selvitä yleismittarin dioditesterillä näytön nastat ennen juottamista verolevylle. Yleensä näytössä on eri puolilla nasta yhteiselle plus- tai miinusnavalle. Löydät sen mittarin dioditesterillä käymällä mittarin antureilla läpi näytön vastakkaisten puolten jalkoja. Mittari piippaa, jos olet löytänyt yhteisen nastan. Tämän jälkeen voit kytkeä mittarin anturin jompaankumpaan yhteisistä nastoista ja käydä toisella anturilla läpi muita nastoja. Mittarin virta riittää syyttämään lediä himmeästi. Jos yksikään ledi ei syty, vaihda mittarin anturit toisin päin.



Kuva 9. Segmenttinäyttö juotettuna paikalleen ja kytkettynä. Tila käy helposti ahtaaksi, ja johdoista tulee spagettia.

Jutun esimerkkilaitte rakennettiin niin, että näytön neljä alinta segmenttiä vastaavat tikun suuntia ja piste tulituspappia.

Jos käytössäsi ei ole yleismittaria, voit selvittää näytön toiminnallisuuden kytkemällä näytön kaikki nastat piirilevylle. Kytke syöttöjännitteestä vastuksen kautta irrallinen johdonpää sekä toinen johto maasta ja kokeile näillä johdoilla näytön jalkoja läpi. Etuvastuksen ansiosta et voi rikkoa näyttöä. Kirjoita ylös, mikä nasta vastaa mitään segmenttiä.

Vaihe 5a (jos käytät näyttöä)

Kytke näytön yhteinen anodi tai katodi.

Voit asettaa näytön verolevylle, juottaa nastat ja kytkeä näytön yhteisen nastan: anodin virransyöttöön tai katodin maahan. Juota myös etuvastus anodin ja ledin väliin.

Vaihe 5b (jos et käytä näyttöä)

Kytke kunkin ledin miinus vastuksen kautta maahan.

Vaihe 6

Kytke D-liittimen nastat verolevyltä näyttöön tai ledeihin.

Termit

- LED – *Light Emitting Diode*, valodiode.
- Vastus – *Resistor*, komponentti joka rajoittaa virran määrää. Yksikkö ohmi (Ω).
- Verolevy – Prototyyppien tekemiseen tarkoitettu piirilevy, jolla on ristikkossa reikiä. Toisella puolella reikien ympärillä on kuparia. Komponenttien jalat voi työntää rei'istä läpi ja juottaa siten toisen puolen kuparireunukseen.

Vedä D-liittimen nastoista hyppylan- goilla erillisten ledien miinusnapoihin tai vaiheessa 5a tekemäsi suunnitel- man mukaisesti näytön segmenttien nastoihin.

Ensimmäiseksi kannattaa juottaa paikalleen maa (nasta 8) ja sen jälkeen haluamassasi järjestyksessä suunnat ja tulitus. Kokeille testeriasi aina juotok- sen tekemisen jälkeen ehjän joystickin kanssa ja varmista, että se toimii odo- tetusti.

Vaihe 7

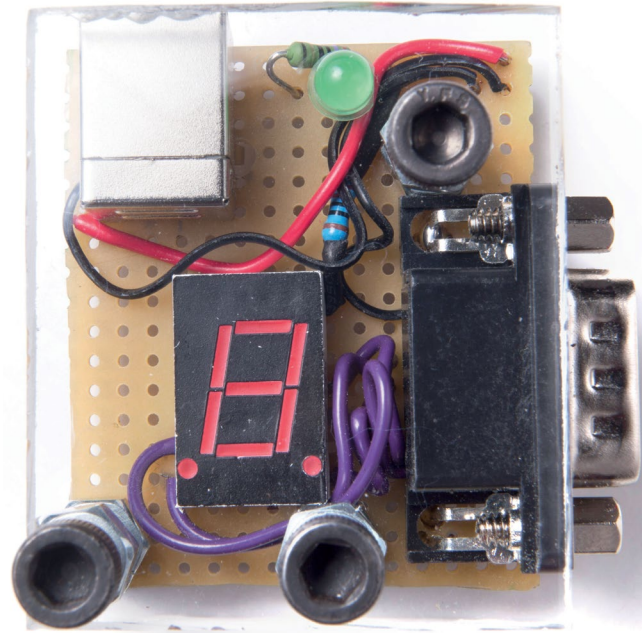
Tarkista vielä, ettei missään ole siltauk- sia. Kytke virrat laitteeseen ja testaa toimivaksi tiedetyllä joystickilla, että kustakin liikkeestä syytyy oikea ledi. Korjaa mahdolliset väärin kytketyt suunnat.

Viimeistely

Viimeistään tässä vaiheessa voit sahata verolevystä ylimääräiset osat pois niin, että saat piirilevystä mahdollisim- man pienen. Jos käytät johdonpätkien varassa olevaa D-liittintä, kannattaa keksiä jokin muovirasia, johon saat

testerisi laitettua niin, että D-liittimen saa kiinnitettyä tukevasti paikalleen. Jos D- liittin on piirilevyn reunassa juotettuna, riittää kevyempi kotelointi. Itse tein läpinä- kyvästä akryylilevystä piiri- levyn kokoiset levyt, jotka kiinnitin pulteilla piirilevyn kummallekin puolelle. Hi- fistelijä voi vaikkapa 3D- tulostaa itselleen hienon kotelon.

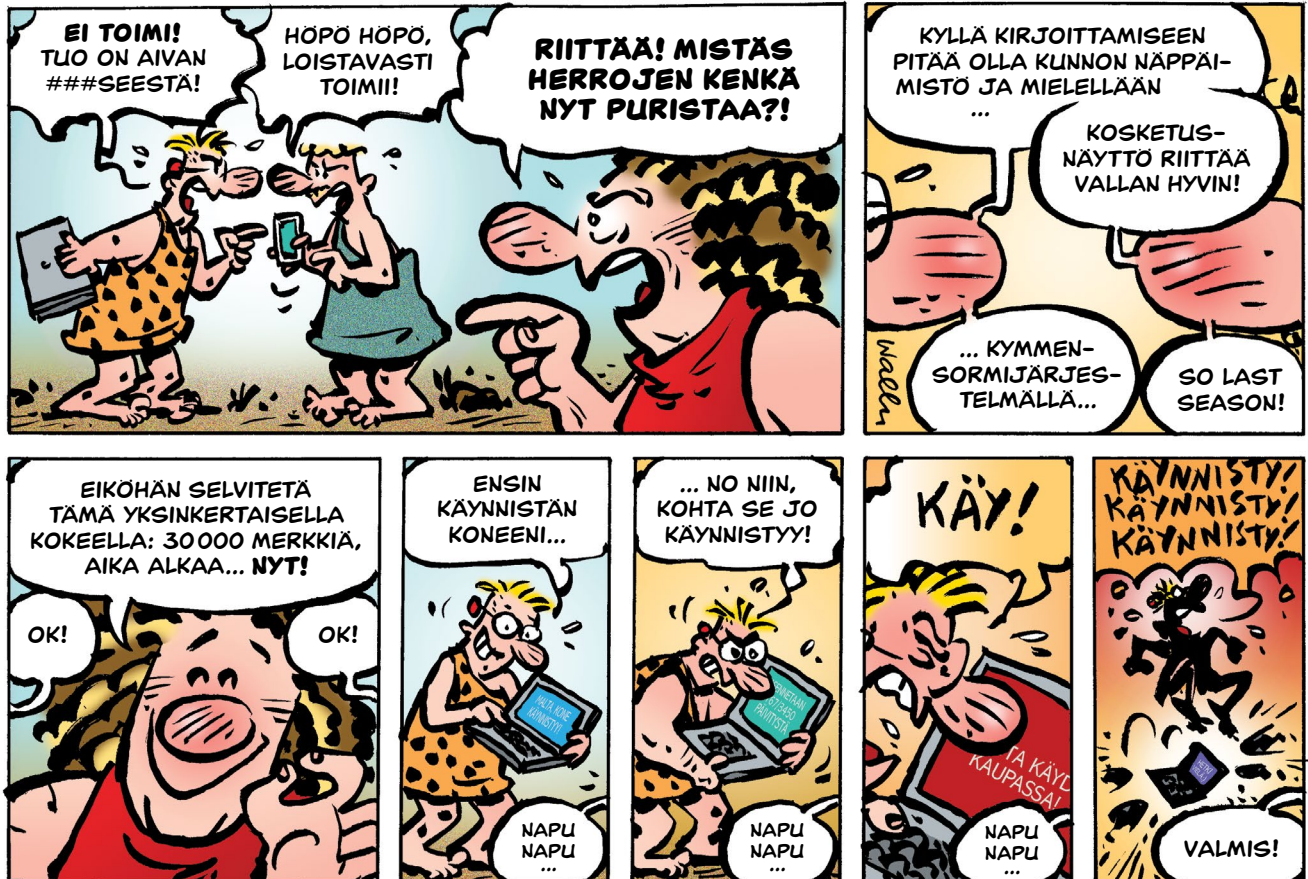
Lopputuloksena syntyi toimiva joystick-testeri. Tes- tasin sitä sekä uuden mikro- kytkimillä tehdyn joystickin kanssa että vanhan, huolta- mattoman TAC-2:n kanssa. Uuden tikun kanssa ledit palavat kirkkaina joka suun- taan, mutta TAC-2:lla ne pa- lavat selvästi himmeämmin ja epätasaisemmin, koska ti- kun sisäiset kontaktipinnat ovat hapet- tuneet. Eli pelkän johdotuksen lisäksi testerillä voi testata tikun kontaktien kuntoa. 🍄



Kuva 10. Lopullinen laite kasattuna.

MIKROKIVIKALSI

TEKSTIPAINOTTEINEN SARJAKLIVA





Itella Green

Are you ready for The Hunt?

68 75 6E

74 2E

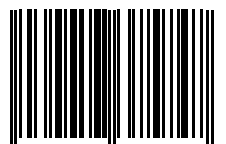
72 69 2E

66 69

Reaktor

Helsinki / Tokyo / New York

reaktor.com/careers



6 430057 375747